



UNIVERSIDADE FEDERAL DO MARANHÃO  
ENGENHARIA DA COMPUTAÇÃO  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA - CCET

Ricardo Kenji Ramos Tanabe

# **Segmentação Automática de Disco Óptico em Imagens de Retina**

São Luís - MA

2023

Ricardo Kenji Ramos Tanabe

# **Segmentação Automática de Disco Óptico em Imagens de Retina**

Trabalho de Contextualização e Integração Curricular, apresentado ao curso de Engenharia da Computação, como requisito para a obtenção do Título de Bacharel em Engenharia da Computação. Centro de Ciências Exatas e Tecnológicas da Universidade Federal do Maranhão.

Orientador: Prof. Dr. Bruno Feres de Souza

São Luís - MA

2023

Ricardo Kenji Ramos Tanabe

## **Segmentação Automática de Disco Óptico em Imagens de Retina**

Trabalho de Contextualização e Integração Curricular, apresentado ao curso de Engenharia da Computação, como requisito para a obtenção do Título de Bacharel em Engenharia da Computação. Centro de Ciências Exatas e Tecnológicas da Universidade Federal do Maranhão.

Trabalho \_\_\_\_\_ em São Luís - MA, 18 de dezembro de 2023:

---

**Prof. Dr. Bruno Feres de Souza**  
Orientador

---

**Prof. Dr. Haroldo Gomes Barroso  
Filho**  
Examinador

---

**Prof. Dr. Paulo Rogério de Almeida  
Ribeiro**  
Examinador

São Luís - MA  
2023

# Agradecimentos

Gostaria de expressar minha sincera gratidão a todas as pessoas e instituições que contribuíram para a conclusão deste Trabalho de Conclusão de Curso. Este projeto representou um marco significativo em minha jornada acadêmica, e não teria sido possível sem o apoio generoso e a orientação de muitos.

Primeiramente, desejo agradecer ao meu orientador, Prof. Dr. Bruno Feres de Souza, pela sua dedicação, paciência e orientação valiosa ao longo de todo o processo. Seu conhecimento foi fundamental para a condução desta pesquisa, e sou imensamente grato pela oportunidade de aprender com sua experiência.

À Universidade Federal do Maranhão - UFMA, expresso meu reconhecimento pela excelência acadêmica proporcionada, pelos recursos disponíveis e pelo ambiente que contribuíram para o desenvolvimento deste trabalho. Cada aula, palestra e interação com colegas moldaram positivamente minha trajetória acadêmica.

Quero estender minha gratidão aos professores Prof. Dr. Haroldo Gomes Barroso Filho e Prof. Dr. Paulo Rogério de Almeida Ribeiro, cujo apoio e feedback construtivo enriqueceram o desenvolvimento deste trabalho. Suas sugestões foram valiosas e contribuíram para a qualidade do resultado final.

À minha família e amigos, agradeço por seu amor, paciência e compreensão ao longo de todo o processo. Seu apoio emocional serviu de base para superar desafios e persistir na busca por excelência acadêmica.

Por fim, aos colegas de classe e colaboradores que compartilharam ideias e experiências, agradeço pela colaboração e pelo ambiente acadêmico enriquecedor que tornou esta jornada mais significativa.

Cada um de vocês desempenhou um papel importante no sucesso deste projeto, e sou profundamente grato por seu apoio. Este TCC representa não apenas uma conquista acadêmica, mas também uma celebração das conexões humanas que enriquecem nosso caminho educacional.

Muito obrigado.

Ricardo Kenji Ramos Tanabe

*"A inovação muitas vezes é a  
capacidade de alcançar a simplicidade,  
e a simplicidade é a  
verdadeira essência da grandeza."*

atribuída a J.C. Penney

# Resumo

A análise de imagens médicas assistida por computador está revolucionando a área da medicina. A introdução desses sistemas aumenta a precisão e a eficiência dos diagnósticos realizados pelos profissionais da saúde. Nesse contexto, diversos métodos de Processamento Digital de Imagens foram desenvolvidos ao longo dos anos para atender a demanda crescente. Diante da diversidade de técnicas, a inteligência artificial tem ganhado notoriedade na execução dessa atividade, em especial, as redes neurais convolucionais.

Assim, este trabalho visa analisar e compreender os processos que regem a localização do disco óptico, elemento que faz parte da estrutura da retina, em imagens digitais médicas, com a finalidade de implementar algoritmo capaz de localizar o objeto da pesquisa. Para isso, realizou-se estudo sobre o tema e direcionou-se para o modelo de rede neural convolucional denominado U-Net.

**Palavras-chave:** Disco Óptico, Processamento Digital de Imagens, U-Net.

# Abstract

The computer-assisted analysis of medical images has been revolutionizing the field of medicine. The introduction of these systems enhances the accuracy and efficiency of diagnoses performed by healthcare professionals. In this context, various methods of Digital Image Processing have been developed over the years to meet the growing demand. Given the diversity of techniques, artificial intelligence has been gaining prominence in the execution of this activity, especially convolutional neural networks.

Therefore, this work aims to analyze and understand the processes that govern the localization of the optic disc, an element that is part of the retina structure, in medical digital images, with the purpose of implementing an algorithm capable of locating the object of the research. To achieve this, a study on the topic was conducted, and it pointed to the convolutional neural network model known as U-Net.

**Keywords:** Optical Disc, Digital Image Processing, U-Net.

# Lista de ilustrações

Figura 1 – Imagem da retina e suas principais estruturas. . . . .	13
Figura 2 – Anatomia básica do olho humano. . . . .	18
Figura 3 – Anatomia da retina. . . . .	19
Figura 4 – Passos fundamentais em PDI. . . . .	25
Figura 5 – <i>Perceptron</i> . . . . .	29
Figura 6 – Rede de <i>Perceptrons</i> . . . . .	30
Figura 7 – Neurônio Sigmoid. . . . .	32
Figura 8 – Função Sigmoid . . . . .	32
Figura 9 – Arquitetura de rede neural . . . . .	33
Figura 10 – Arquitetura de rede neural multicamadas . . . . .	33
Figura 11 – Neurônios de Entrada . . . . .	37
Figura 12 – Campo Receptivo Local . . . . .	38
Figura 13 – Ligação entre a camada de entrada e a camada oculta utilizando um campo receptivo local 5x5 . . . . .	38
Figura 14 – Divisão em mapas de recursos . . . . .	39
Figura 15 – Extraíndo o valor mais significativo na operação max-pooling . . . . .	40
Figura 16 – Rede Neural Convolutacional . . . . .	41
Figura 17 – O modelo U-Net . . . . .	42
Figura 18 – Da esquerda para a direita, começando de cima e indo para baixo, temos: (1) Imagem colorida da retina; (2) Imagem binária do contorno do DO e o ponto é a localização da fóvea; (3) Imagem binária dos vasos sanguíneos extraídos por profissional da área; (4) Imagem binária dos vasos sanguíneos corrigidos. . . . .	52
Figura 19 – Resultado das operações de data augmentation . . . . .	58
Figura 20 – Utilização do matplotlib no ambiente Colaboratory da Google . . . . .	59
Figura 21 – Da esquerda para a direita, a imagem da retina, a máscara predita pelo modelo e a máscara real . . . . .	62
Figura 22 – Resultado da segmentação utilizando as máscaras preditas . . . . .	63
Figura 23 – Avaliação do modelo . . . . .	63



# Lista de tabelas

Tabela 1 – Descrição das estruturas do olho humano. . . . .	17
Tabela 2 – Descrição das estruturas da retina. . . . .	19
Tabela 3 – Resultado do modelo resumido . . . . .	56
Tabela 4 – Resultado do Treinamento . . . . .	61

# Lista de abreviaturas e siglas

UFMA	<i>Universidade Federal do Maranhão</i>
PDI	<i>Processamento Digital de Imagens</i>
CAD	<i>Computer Aided Diagnosis</i>
DO	<i>Disco Óptico</i>
DMRI	<i>Degeneração Macular Relacionada a Idade</i>
IA	<i>Inteligência Artificial</i>
MLP	<i>Multilayer Perceptrons</i>
ReLU	<i>Rectified Linear Unit</i>
AdaGrad	<i>Adaptive Gradient</i>
RMSProp	<i>Root Mean Square Propagation</i>

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
<b>1.1</b>	<b>Justificativa</b>	<b>14</b>
<b>1.2</b>	<b>Objetivos</b>	<b>15</b>
1.2.1	Geral	15
1.2.2	Específicos	15
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>16</b>
<b>2.1</b>	<b>Sob a ótica dos profissionais da saúde</b>	<b>16</b>
2.1.1	Anatomia básica do olho humano	17
2.1.1.1	A retina e seus elementos	18
2.1.2	Patologias relacionadas à visão	19
2.1.2.1	Degeneração macular relacionada a idade (DMRI)	20
2.1.2.2	Retinopatia diabética	20
2.1.2.3	Glaucoma	20
2.1.3	Resumo	21
<b>2.2</b>	<b>Técnicas de Computação no Tratamento de Imagens</b>	<b>21</b>
2.2.1	Sob a ótica da disciplina Processamento Digital de Imagens (PDI)	22
2.2.1.1	Princípios fundamentais em processamento de imagens	23
2.2.1.2	Segmentação de imagens	24
2.2.2	Sob a ótica da disciplina de inteligência artificial	26
2.2.2.1	<i>Machine Learning</i>	27
2.2.2.2	<i>Deep Learning</i>	28
2.2.2.3	Redes Neurais	28
2.2.2.3.1	<i>Perceptrons</i>	29
2.2.2.3.2	Neurônios sigmoid	31
2.2.2.3.3	A arquitetura das redes neurais	32
2.2.2.3.4	Algoritmos de retropropagação ( <i>backpropagation</i> ) e descida do gradiente	34
2.2.2.4	Redes Neurais Convolucionais	36
2.2.2.5	U-Net	41
2.2.2.6	Funções de Perda e Métricas de Desempenho	43
2.2.2.6.1	Dice Coefficient e <i>Dice Loss</i>	44
2.2.2.6.2	<i>IoU</i>	45
2.2.2.7	Algoritmos de Otimização	45
2.2.2.7.1	Otimizador Adam	47
2.2.2.8	Técnica para aumentar a quantidade de dados	48

<b>3</b>	<b>METODOLOGIA DA PESQUISA</b>	<b>50</b>
3.1	Descrição do Modelo U-Net	50
3.2	Pré-processamento de Dados	50
3.3	Conjunto de Dados	51
3.4	Treinamento do Modelo	51
3.5	Avaliação do Modelo	51
3.6	Ferramentas e Ambiente de Desenvolvimento	52
3.7	Validação e Teste	53
3.8	Considerações finais sobre a metodologia	53
<b>4</b>	<b>DESENVOLVIMENTO</b>	<b>54</b>
4.1	Implementação do Modelo	54
4.2	Detalhes Técnicos	56
4.3	Tratamento de Dados	57
4.4	Ferramentas e Bibliotecas	57
4.5	Ambiente de Desenvolvimento	59
4.6	Testes e Validação	60
4.7	Problemas e Soluções	60
4.8	Resultados Intermediários	61
4.9	Iterações e Refinamentos	61
4.10	Conexão com a Metodologia	62
<b>5</b>	<b>DISCUSSÃO</b>	<b>64</b>
5.1	Avaliação do Desempenho do Modelo	64
5.2	Identificação de Problemas	64
5.3	Interpretação de Resultados	64
5.4	Análise Visual	64
5.5	Possíveis Melhorias	65
5.6	Impacto nos Objetivos do Projeto	65
<b>6</b>	<b>CONCLUSÃO</b>	<b>66</b>
	<b>REFERÊNCIAS</b>	<b>67</b>

# 1 Introdução

A década de 1940 marcou um período chave para o surgimento das bases do que conhecemos hoje como o computador digital moderno. A introdução do modelo computacional proposto por John von Neumann e uma série de avanços significativos, como o desenvolvimento nas áreas de armazenamento em massa e de sistemas de visualização, proporcionaram aos computadores os recursos necessários para ingressarem no campo do processamento digital de imagens (PDI). As técnicas de PDI emergiram com o desenvolvimento contínuo dos computadores, encontrando aplicações em diversas áreas, incluindo a análise de imagens médicas ([GONZALEZ; WOORS, 2010](#)).

Esses eventos convergiram para o desenvolvimento de sistemas automáticos de diagnóstico médico, formalmente denominados como Sistemas de Diagnóstico Auxiliado por Computador (CAD – *Computer Aided Diagnosis*). Os Sistemas CAD desempenham um papel fundamental na detecção automatizada de anomalias em exames, proporcionando agilidade e aprimorando a precisão nos diagnósticos. O procedimento engloba a análise de imagens médicas e a aplicação de técnicas de reconhecimento de padrões ([PIXEON, 2014](#)). As áreas médicas que se destacam na aplicação do CAD incluem o uso de sistemas automatizados em mamografia, tomografia computadorizada de tórax e ressonância magnética ([CLARO et al., 2018](#)).

O campo de aplicação do CAD é vasto, e destaca-se na Oftalmologia como um terreno fértil. Nesse contexto, diversas patologias, desde condições graves como diabetes e hipertensão até outras relacionadas a alterações vasculares, sanguíneas e nervosas, além de uma variedade de doenças oculares, podem ser diagnosticadas e avaliadas por meio da análise da retina. Técnicas como mapeamento da retina, oftalmoscopia direta ou indireta, entre outras, desempenham um papel fundamental nesse processo diagnóstico ([HOS, 2016](#)) ([RIORDAN-EVA; WHITCHER, 2011](#)).

Conforme os dados da pesquisa Vigitel 2019 (Vigilância de Fatores de Risco e Proteção para Doenças Crônicas por Inquérito Telefônico) ([MINISTERIO DA SAUDE, 2020](#)), foi traçado o perfil epidemiológico do brasileiro em relação às doenças crônicas mais prevalentes no país, Os resultados indicam que 7,4% da população apresenta diabetes, 24,5% hipertensão e 20,3% está classificada como obesa.

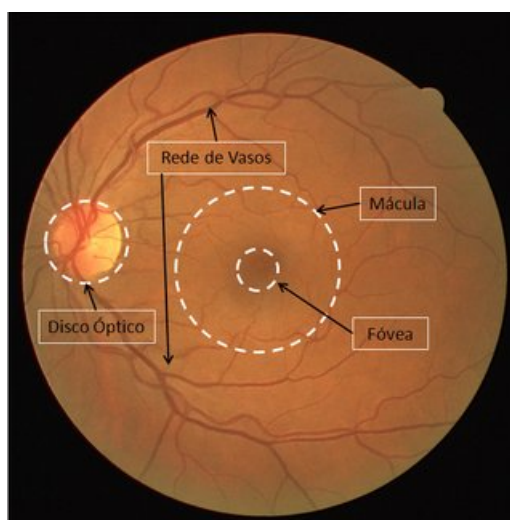
De acordo com a pesquisa, no período de 2006 a 2019, observou-se um aumento na prevalência de diabetes, passando de 5,5% para 7,4%, e da hipertensão arterial, que subiu de 22,6% para 24,5%. Uma análise dos Arquivos Brasileiros de Cardiologia ([SOCIEDADE BRASILEIRA DE CARDIOLOGIA, 2016](#)) revela que, no Brasil, a hipertensão arterial afeta 32,5% (36 milhões) dos adultos, atingindo mais de 60% dos idosos e contribuindo

direta ou indiretamente para 50% das mortes por doença cardiovascular.

A importância da pesquisa em novos sistemas CAD relacionados a doenças oculares é evidenciada pela extensa literatura existente, composta por diversos artigos e estudos publicados nas últimas décadas (CARDOSO, 2019) (SALLOW; SAHIKHA, 2019) (SAYADIA et al., 2018) (PRAKASH; GR; M, 2016). Cada investigação contribui de maneira única para o conhecimento do tema. Indivíduos afetados por doenças oculares muitas vezes desconhecem a progressão assintomática até atingir estágios avançados, nos quais o tratamento se torna menos eficaz na prevenção da deficiência. Assim, o desenvolvimento desses sistemas não apenas auxilia na prevenção das doenças oculares, mas também alivia a carga sobre os profissionais da saúde que atuam nessa área (ZHANG et al., 2014).

Uma imagem típica da retina apresenta elementos essenciais, tais como a Mácula, Fóvea, Disco Óptico e Vasos Sanguíneos (Veia e Arteriola). Entre esses elementos, a identificação do Disco Óptico (DO) assume um papel importante em muitos sistemas CAD. O DO serve como ponto de referência inicial para deduzir a localização dos demais componentes estruturais da retina, desempenhando também um papel fundamental na detecção de condições oftalmológicas como retinopatia diabética, glaucoma e degeneração macular relacionada à idade (ERWIN; CANIA; LARASATI, 2020).

Figura 1 – Imagem da retina e suas principais estruturas.



Fonte: Métodos computacionais para segmentação do disco óptico em imagens de retina: uma revisão (CLARO et al., 2018).

O presente trabalho concentra-se na análise de uma das estruturas fundamentais da retina: o disco óptico (DO). A proposta visa explorar a literatura e selecionar técnica empregada na localização do DO, com o objetivo de desenvolver um algoritmo capaz de identificar seu contorno. Esse algoritmo é projetado para trabalhar tanto com imagens de retinas saudáveis quanto com imagens de retinas que apresentem patologias. O conjunto de

dados utilizado encontra-se disponível no site pessoal do Farnell, denominado *Automatic Retinal Image Analysis (ARIA) Data Set* (FARNELL, 2020).

O planejamento adotado para a realização da pesquisa bibliográfica seguiu o seguinte cronograma: (1) compreensão das características essenciais do olho humano e da retina, considerando seus elementos; (2) exploração dos conteúdos relevantes na área de Processamento Digital de Imagens e das descobertas mais recentes, visando a resolução do problema.

Ao explorar o tema, procurou-se compreender, sob a perspectiva médica, os aspectos relevantes relacionados à pesquisa realizada. Como mencionado no primeiro item acima, constatou-se que a compreensão dos elementos que constituem a visão humana e a retina é condição necessária e suficiente para alcançar a meta proposta.

A etapa central da pesquisa concentra-se no Processamento Digital de Imagens. Ao longo das décadas, cientistas e pesquisadores têm desenvolvido técnicas cada vez mais sofisticadas e refinadas para o tratamento de imagens em diversos contextos. Atualmente, com o avanço contínuo de computadores dotados de elevado poder de processamento, a atenção volta-se para os estudos relacionados a um dos modelos de Rede Neural Convolutacional, conhecido como U-Net.

Assim, o presente trabalho foi dividido em referencial teórico onde organizou-se em dois tópicos centrais, cada um desenvolvendo as ideias relevantes a área de interesse; metodologia no qual apresenta o modelo U-Net em conjunto com os algoritmos de função de perda, de métrica de desempenho e de otimização selecionados; desenvolvimento que apresenta a implementação do modelo; discussão no qual verifica-se os problemas e a validade do modelo e; conclusão.

## 1.1 Justificativa

O globo ocular contém elementos que estão intimamente ligados a saúde geral do organismo. Os métodos de prevenção não são invasivos, além de permitirem a detecção de diversas doenças que possam acometer o indivíduo. O ponto de partida é a localização do DO, no qual é referência para a localização dos demais elementos de igual importância, pois possibilita a detecção de outras mazelas.

Automatizando o procedimento de análise da retina, uma carga significativa de trabalho é retirada dos profissionais que atuam na área, além de permitir que uma quantidade maior de pessoas possa ser analisada e diagnosticada, aumentando sua qualidade de vida.

Diversas pesquisas sobre o tema estão sendo realizadas e divulgadas a fim de alcançar a eficiência necessária para tornar possível a automatização dos diagnósticos de

análise da retina, com o intuito de incorporar como um componente padrão no dia a dia dos profissionais da saúde.

Assim, este trabalho pretende contribuir com o desenvolvimento de sistemas CAD relacionados a segmentação de imagens da retina com o objetivo de localizar o disco óptico.

## 1.2 Objetivos

### 1.2.1 Geral

Analisar e compreender os processos que regem a localização do DO, com base na literatura disponível, visando a implementação de um sistema que utiliza técnicas de processamento de imagens capaz de localizar o objeto da pesquisa com índices de precisão satisfatórios e de forma automatizada.

### 1.2.2 Específicos

- Entender os componentes fundamentais da fisiologia do olho necessários para o tema;
- Analisar literatura disponível sobre técnicas de processamento digital de imagens aplicadas em imagens de retina;
- Identificar e selecionar um conjunto de técnicas de segmentação do disco óptico;
- Avaliar a possibilidade de incorporação das técnicas;
- Projetar algoritmos para a localização do disco óptico.



## 2 Referencial Teórico

A interseção entre saúde e a tecnologia tem desencadeado avanços notáveis na compreensão, diagnóstico e tratamento de doenças. No centro dessa convergência encontra-se a análise de imagens médicas, uma disciplina que requer profissionais qualificados em conjunto com técnicas de computação avançadas. Assim, este referencial teórico foi dividido em dois temas centrais: (1) Sob a ótica dos profissionais da saúde e (2) Técnicas de computação no tratamento de imagens.

### 2.1 Sob a ótica dos profissionais da saúde

Conforme afirmado por ([RIORDAN-EVA; WHITCHER, 2011](#)), uma compreensão abrangente da anatomia do olho, da órbita, das vias visuais, dos nervos cranianos superiores e das vias centrais para o controle dos movimentos oculares é essencial para uma interpretação adequada das doenças com manifestações oculares.

Ao considerar todos os órgãos do corpo, o olho destaca-se como o mais acessível ao exame direto. Possui a singularidade de ser a única parte do corpo em que os vasos sanguíneos e tecidos do sistema nervoso central (retina e nervo óptico) podem ser visualizados diretamente via exames. Conseqüentemente, diversas doenças oculares podem ser identificadas por meio deste exame.

Segundo ([AZEVEDO-MARQUES et al., 2018](#)), a retina é a única área que permite uma observação direta e não invasiva dos vasos sanguíneos. A Fundoscopia, também conhecida como exame de fundo de olho, oftalmoscopia ou mapeamento da retina, é empregada para diagnosticar diversas condições, incluindo glaucoma, retinopatia diabética e degeneração macular relacionada à idade. Além disso, esse exame não se limita ao diagnóstico de doenças oftalmológicas, abrangendo também a detecção de condições sistêmicas como hipertensão arterial e possibilitando a identificação de eventos críticos, como hemorragia cerebral, hemorragia subaracnóidea e encefalopatia hipertensiva por meio da arteriosclerose retiniana.

Diante dos fatos apresentados, é razoável admitir a importância desses diagnósticos, considerando a riqueza de informações úteis obtidas por meio do exame da retina. Nesse contexto, destaca-se a estrutura do olho humano e, em particular, a retina, objeto de análise neste estudo.

### 2.1.1 Anatomia básica do olho humano

Considerando os cinco sentidos, a visão é provavelmente a mais complexa. O olho humano é formado por três camadas distintas: (1) a camada mais externa, composta pela esclera e pela córnea; (2) a camada média, formada pelo coróide, corpo ciliar e íris; e (3) a camada mais interna, que consiste na retina (SIEGFRIED, 2010).

Apesar da existência de detalhes minuciosos, para os propósitos desta pesquisa, é suficiente uma breve descrição de cada estrutura, a fim de compreender os elementos que o compõem e suas funções. (LAVINNE; M.D., 2001) fornece as informações fundamentais sobre os componentes, resumidos na Tabela 1:

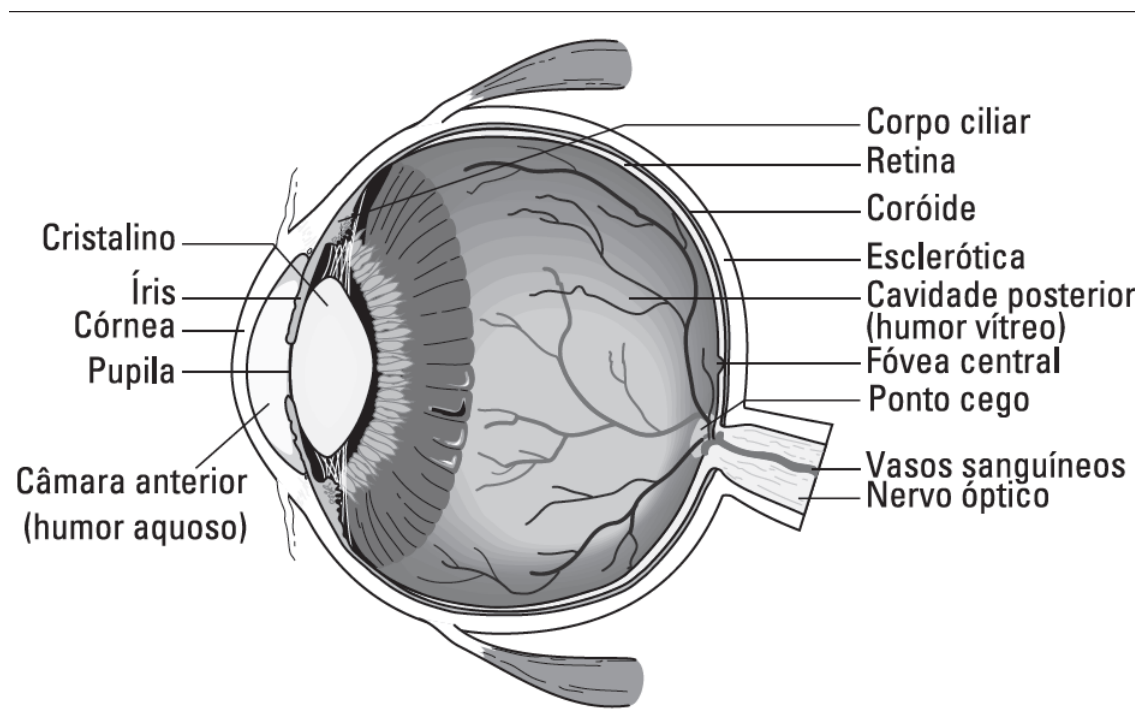
<b>Estrutura do olho humano</b>	<b>Descrição</b>
Pupila	Geralmente, é uma estrutura de coloração preta responsável por regular a entrada de luz no sistema visual.
Íris	Membrana colorida do olho que controla a dilatação da pupila por meio de músculos presentes na estrutura.
Córnea	Camada que serve como lente, direcionando os raios de luz para a retina. Também protege a íris e a pupila contra infecções e corpos estranhos.
Cristalino	Localizado na parte posterior da pupila, atua como uma lente, focalizando os raios de luz que chegam na retina para que a imagem possa ser vista claramente.
Humor vítreo	Substância transparente e gelatinosa que preenche o espaço vítreo, posterior ao cristalino. Responsável pelo formato arredondado do olho e permite a entrada de luz até o fundo do globo ocular.
Retina	Localizado na parte posterior do globo ocular, é a membrana mais interna da estrutura, responsável por formar imagens e sendo o tecido mais complexo do olho. A retina contém dois tipos de fotorreceptores: (1) Bastonetes, que auxiliam na visão noturna e são sensíveis a movimentos; (2) Cones, que detectam cores e detalhes mínimos. Existem três tipos de cones, cada um responsável por detectar uma das cores: vermelha, verde e azul. Sua ausência ou danificação resulta em cegueira para as cores.
Nervo óptico	Tem a função de conectar o cérebro aos nossos olhos, agindo como um condutor. Os impulsos nervosos são gerados quando a luz excita os bastonetes e cones, sendo enviados às células que formam o nervo óptico.

Tabela 1 – Descrição das estruturas do olho humano.

A Fig. 2 mapeia os elementos que compõem o olho humano.

A camada de interesse nesta pesquisa é a retina, que será discutida a seguir.

Figura 2 – Anatomia básica do olho humano.



Fonte: Anatomia e Fisiologia para Leigos ([SIEGFRIED, 2010](#))

#### 2.1.1.1 A retina e seus elementos

Segundo ([BOWLING, 2016](#)), a retina humana é uma estrutura altamente organizada, tendo a capacidade de dar início ao processamento visual antes que ela seja transmitida através do nervo óptico para o córtex visual, região onde ocorre o processamento das informações visuais.

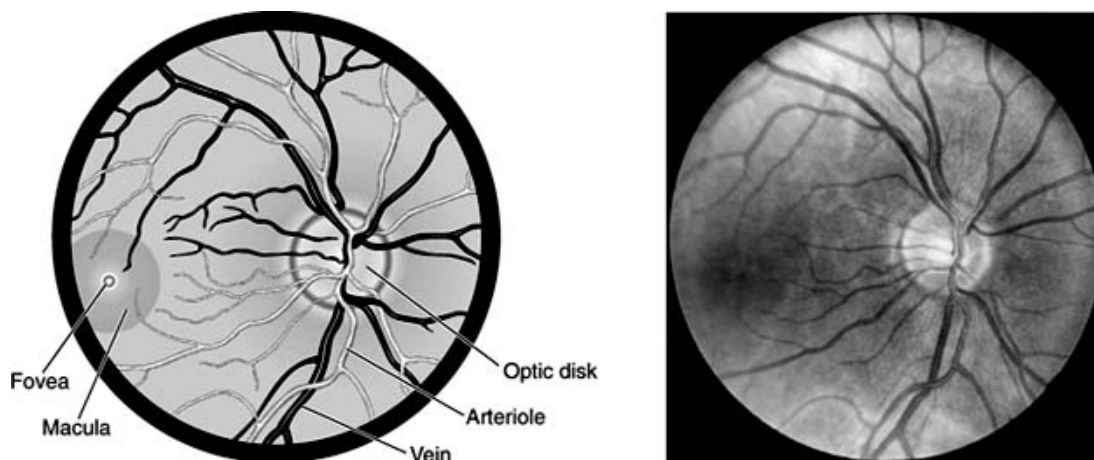
A Fig. 3 mapeia os elementos que compõem a retina e a Tabela 2 apresenta uma breve descrição das estruturas mais proeminentes, visualmente, da retina: Disco Óptico; Arteriola; Veias; Fóvea e; Macula:

No exame de fundo de olho, realizado na oftalmoscopia direta, no qual se obtém uma imagem monocular, incluindo uma visão do fundo de olho aumentada em 15 vezes, conforme ([RIORDAN-EVA; WHITCHER, 2011](#)), exige, dentre outras atividades, detalhes da retina visando o foco nítido. Considerando que os vasos da retina surgem a partir do disco, infere-se em uma primeira observação que a detecção adequada do disco óptico auxilia à localização dos demais componentes da estrutura. A partir do disco óptico, é necessário o estudo da forma, o tamanho e a cor do disco, a nitidez de suas margens, bem como a dimensão da “escavação fisiológica” central pálida. A razão entre o tamanho da escavação e o tamanho do disco tem importância diagnóstica no glaucoma.

<b>Estrutura da retina</b>	<b>Descrição</b>
Disco Óptico	É um disco de formato oval localizado na parte de trás do olho e é composto por diversas células nervosas, estas denominadas células ganglionares. Sua principal função é captar informações provindas dos cones e bastonetes.
Arteriola	São vasos sanguíneos de pequena dimensão que resultam de ramificações das artérias. Regulam principalmente a resistência ao fluxo sanguíneo e, portanto, a pressão sanguínea periférica.
Veias	São vasos sanguíneos cuja função é garantir que o sangue, presente em vários tecidos do corpo, retorne ao coração.
Fóvea	Também conhecido como mancha amarela, é a região central do olho humano onde se concentram os cones e onde se forma a imagem que será transmitida ao cérebro.
Mácula	Tem como principal função garantir uma visão nítida e clara dos objetos em nosso campo de visão. Também é responsável por toda a visão central e por parte da percepção das cores.

Tabela 2 – Descrição das estruturas da retina.

Figura 3 – Anatomia da retina.



Fonte: Chapter 2: Ophthalmologic Examination ([MCGRAW-HILL COMPANIES, 2023](#)).

### 2.1.2 Patologias relacionadas à visão

A maioria das patologias relacionadas à visão podem causar a diminuição da capacidade visual, desconforto psicológicos e físicos e, em casos mais graves, causar a perda definitiva da visão. Essas patologias normalmente surgem por alguma condição hereditária familiar ou com o processo de envelhecimento ([VIVA OFTALMO, 2020](#)).

Diante da realidade apresentada é importante ressaltar que: a maioria das doenças relacionadas à visão, quando diagnosticadas cedo e de maneira correta, podem ser prevenidas e tratadas.

Dentre as doenças oculares relacionadas à retina, temos: Degeneração macular relacionado à idade; Retinopatia diabética; Glaucoma, apresentadas a seguir.

#### 2.1.2.1 Degeneração macular relacionada a idade (DMRI)

A DRMI é a principal causa de cegueira irreversível nos países desenvolvidos. Trata-se de uma doença progressiva multifatorial complexa com influências genéticas e ambientais. Pessoas acima de 55 anos de idade são o grupo de risco para a DRMI.

Apesar de ser pouco compreendida, existem pistas sobre os elementos essenciais ligados à doença, como a degeneração do epitélio pigmentário da retina, ligada ao estresse oxidativo. Mudanças na matriz extracelular adjacente da *Membrana de Bruch* e a formação de depósitos sub-retinianos são fundamentais para a evolução da doença. A perda visual resulta da progressão degenerativa até a morte celular e atrofia do epitélio pigmentário da retina.

#### 2.1.2.2 Retinopatia diabética

Considerado um dos principais agentes responsáveis por causar cegueira no ocidente, especialmente entre indivíduos em idade reprodutiva. Grupo de pessoas diabéticas são propensas a apresentar a patologia além dos fatores de risco, tais como a hiperglicemia crônica; hipertensão arterial; hipercolesterolemia e o tabagismo aumentam a chance no desenvolvimento e progressão da doença.

O diagnóstico e tratamento precoce da retinopatia diabética são essenciais. Comprovou-se que a fotografia digital do fundo de olho é um método sensível e eficaz para a triagem.

#### 2.1.2.3 Glaucoma

Considerado uma neuropatia óptica crônica adquirida, o glaucoma se caracteriza por escavação do disco óptico e perda de campo visual. Normalmente está associado a elevação da pressão intraocular e, na maioria dos casos, não existe doença ocular associada (glaucoma primário).

([RIORDAN-EVA; WHITCHER, 2011](#)) afirma que cerca de 60 milhões de pessoas têm glaucoma. O glaucoma primário de ângulo aberto, forma mais comum ente negros e brancos, causa perda bilateral progressiva assintomática e insidiosa, em geral despercebida até que já tenha ocorrido perda extensa do campo visual. O glaucoma primário de ângulo fechado atinge de 10 a 15% entre os brancos, sendo mais proeminentes entre os asiáticos e inuítes. Na China, o glaucoma primário de ângulo fechado pode ser responsável por mais de 90% da cegueira bilateral decorrente do glaucoma.

O tratamento e sua eficácia, em todos os pacientes com glaucoma, são avaliadas por medição regular da pressão intraocular (tonometria), inspeção dos discos ópticos e exame dos campos visuais. Cabe ressaltar, que a detecção do glaucoma em casos assintomáticos depende da assistência de toda a equipe médica e da cooperação do paciente.

### 2.1.3 Resumo

Ao fim desta leitura, o leitor deve entender que, sob a ótica dos profissionais da saúde, para a adequada interpretação das doenças com manifestações oculares é de suma importância possuir pleno domínio sobre a anatomia do olho e suas estruturas complementares.

Para a anatomia do olho humano, observou-se os elementos que compõem sua estrutura, tais como: Pupila; Íris; Córnea; Cristalino; Humor vítreo; Retina e; Nervo óptico. Dentre elas, a retina é o principal elemento a ser observado para este trabalho, tendo em mente as principais estruturas que a compõem, como: Disco Óptico; Arteriola; Veias; Fóvea e; Mácula.

Por fim, demonstrou-se de forma sucinta três doenças que acometem a população em uma parcela alarmante: o glaucoma, retinopatia diabética, degeneração macular relacionada à idade.

O próximo tópico apresenta as técnicas de computação no tratamento de imagens.

## 2.2 Técnicas de Computação no Tratamento de Imagens

Na computação, a área que trabalha com tratamento de imagens é conhecida como Processamento Digital de Imagens. Dentre os conteúdos apresentados por esta disciplina, temos o campo chamado Segmentação de Imagens.

O interesse nos métodos de Processamento Digital de Imagens surge através de duas áreas principais de aplicação: o aperfeiçoamento das informações visuais para a interpretação humana e o processamento de dados de imagens para armazenamento, transmissão e representação, considerando a percepção automática por máquinas.

Entretanto, considerando as inúmeras variáveis que tornam imagens distintas umas das outras, incluindo imagens que, em primeira análise, aparentam possuir as mesmas características, a técnica clássica de segmentação deve ser ajustada ao problema em questão, tornando, na prática, uma atividade desafiadora.

Outra questão de igual relevância, diz respeito a automação das operações de segmentação, de preferência, com a mínima supervisão humana possível. Na computação, a área que trata sobre agentes inteligentes é conhecida como Inteligência Artificial (IA).

Dentre os ramos da IA, existe o campo denominado Aprendizagem de Máquina (*Machine Learning*) e dentro desse ramo temos a Aprendizagem Profunda (*Deep Learning*).

O interesse nos métodos de *Deep Learning* surge no estudo das redes neurais, em especial, as redes neurais convolucionais e um de seus modelos denominado U-Net, pois a rede é capaz de aprender as características da imagem e extrair informações a partir delas.

Assim, o presente tópico, concentra-se em condensar o conhecimento das áreas descritas acima, verificando as partes mais gerais de forma breve e detalhando as partes específicas de interesse.

### 2.2.1 Sob a ótica da disciplina Processamento Digital de Imagens (PDI)

Para compreender a definição formal de PDI, é essencial explorar um conjunto de ideias fundamentais encontradas na literatura, as quais serão discutidas a seguir, juntamente com o foco de suas análises: a imagem

Em consulta ao ([DICCIONARIO PRIBERAM, 2023](#)) imagem refere-se à “Representação de pessoa ou coisa”. O ([MICHAELIS, 2023](#)) segue uma linha semelhante, definido-a como “Representação do aspecto ou formato de pessoa ou objeto através de desenho, gravura, escultura”. Contudo, para os propósitos do PDI, a definição matemática desempenha um papel crucial, uma vez que os métodos e técnicas aplicados baseiam-se fortemente nas ferramentas fornecidas por essa ciência.

Dessa forma, define-se imagem como um função bidimensional,  $f(x, y)$ , onde  $x$  e  $y$  são coordenadas espaciais (plano), e a amplitude de  $f$  em qualquer par de coordenadas  $(x, y)$  é denominada intensidade ou nível de cinza da imagem nesse ponto ([GONZALEZ; WOORS, 2010](#)).

As imagens digitais são geradas por dispositivos especializados chamados sensores, que, após um complexo processo de captação e interpretação da imagem (não abordado neste trabalho), convertem as funções contínuas em funções discretas. Essa conversão é realizada por meio dos processos de amostragem e quantização. Em resumo, tanto a amostragem quanto a quantização discretizam funções contínuas, diferindo apenas nas variáveis a serem convertidas, sendo, respectivamente, as coordenadas e a amplitude.

A ferramenta central do PDI é o computador digital, responsável por processar a imagem em sua forma digital e/ou para sua forma digital. A tela do computador é plana, e as imagens geradas por ela são representadas por matrizes sobrepostas, cada uma relacionada a uma cor, contendo valores de intensidade com suas respectivas coordenadas. Cada ponto dentro da matriz é igualmente espaçado ao longo dos eixos  $x$  e  $y$ . Esses pontos são conhecidos como elementos pictóricos, elementos de imagem, pels e pixels, sendo ‘pixel’ a terminologia escolhida para o restante do texto.

Com as informações apresentadas, (GONZALEZ; WOORS, 2010) adiciona um paradigma que engloba três tipos de processos computacionais:

- Processos de nível baixo: Incluem operações primitivas, como pré-processamento de imagens para redução de ruído, realce de contraste e nitidez. Tanto a entrada quanto a saída desses processos e operações são imagens.
- Processos de nível médio: Nessa etapa, durante a execução dos processos e operações, as entradas são geralmente imagens, mas as saídas são atributos extraídos delas, como bordas, contornos e identificação de objetos individuais.
- Processos de nível alto: Envolvem a classificação, segundo critérios estabelecidos, de um conjunto de objetos reconhecidos, como na análise de imagens, e realizam funções cognitivas normalmente associadas à visão.

Portanto, o PDI, para os propósitos deste trabalho, abrange processos cujas entradas e saídas são imagens e, adicionalmente, envolve processos de extração de atributos de imagens até - e inclusive - o reconhecimento de objetos individuais.

#### 2.2.1.1 Princípios fundamentais em processamento de imagens

(GONZALEZ; WOORS, 2010) aborda sobre os passos fundamentais em PDI e, para simplificar o entendimento, os métodos de PDI foram divididos em duas categorias principais: (1) métodos cuja entrada e saída são imagens; (2) métodos cujas entradas podem ser imagens, mas cujas saídas são atributos extraídos dessas imagens. Essa categorização visa proporcionar clareza e compreensão das etapas envolvidas, destacando a flexibilidade desses métodos conforme a natureza específica do problema a ser resolvido.

##### **Categoria 1: Métodos com Entrada e Saída de Imagens**

- **Aquisição de imagens:** Este é o ponto de partida, onde a imagem de interesse é capturada para análise posterior.
- **Realce de imagens:** Envolve a manipulação da imagem para torná-la mais adequada a uma aplicação específica. Essas técnicas são orientadas pelo problema em questão, considerando a diversidade de abordagens disponíveis. O realce é subjetivo, pois depende da avaliação do indivíduo em questão, refletindo a natureza personalizada desse processo.
- **Restauração de imagens:** Focado na melhoria visual da imagem, esse processo é objetivo, pois tende a utilizar modelos matemáticos ou probabilísticos para lidar com degradações.



- **Processamento de imagens coloridas:** Concentra-se na visualização e manipulação das cores presentes na imagem.
- **Wavelets e Processamento Multirresolução:** Fundamentais para representar imagens em vários níveis de resolução.
- **Compressão:** Responsável por técnicas de redução do armazenamento necessário para salvar uma imagem ou a largura de banda necessária para transmiti-la.
- **Processamento morfológico:** Lida com ferramentas para extrair componentes de imagens úteis na representação e descrição da forma.

### **Categoria 2: Métodos com Entradas de Imagens e Saídas de Atributos**

- **Segmentação:** Divide uma imagem em partes ou objetos constituintes, uma tarefa desafiadora no PDI.
- **Representação e Descrição:** Partindo dos resultados da segmentação, essa etapa decide como os dados serão representados (fronteiras ou regiões completas) e descreve os dados para enfatizar características de interesse.
- **Reconhecimento:** Atribui rótulos a objetos com base em seus descritores.

Além desses módulos, o conhecimento prévio ou a interação entre a base de conhecimento e os módulos de processamento são importantes para obter resultados significativos no campo do PDI. A sinergia entre esses elementos, seja em processos de nível baixo, médio ou alto, desempenha um papel fundamental em todo o processo, resultando em insights valiosos a partir de dados visuais.

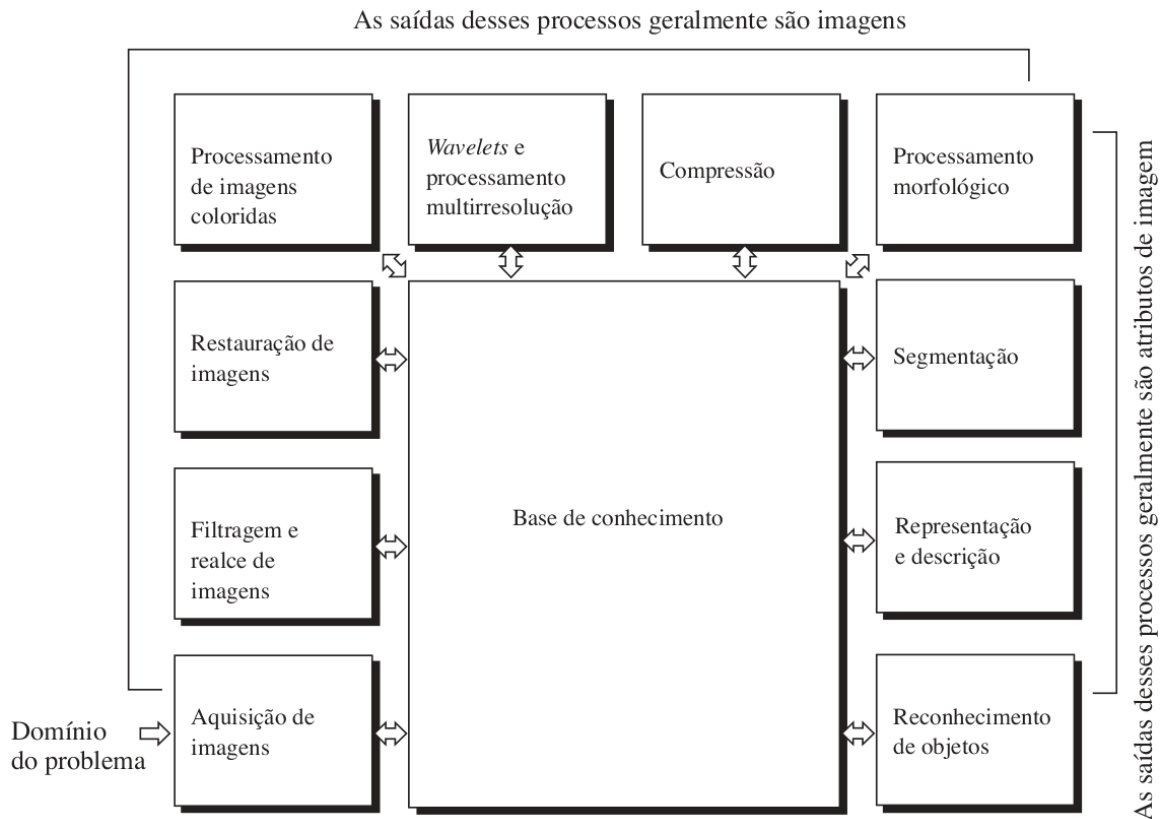
#### 2.2.1.2 Segmentação de imagens

Segundo (SANNIHIT; PURI; SINGH, 2021), segmentação de imagem é o processo de particionar uma imagem em várias partes. Para (GONZALEZ; WOORS, 2010), a segmentação consiste em, dado uma imagem de entrada, receber um atributo ou mais dessa imagem como saída.

Esta técnica subdivide uma imagem em regiões ou objetos que a compõem; deve parar quando os objetos ou as regiões de interesse de uma aplicação forem detectados. Assim, a maioria dos algoritmos de segmentação se baseia nas propriedades básicas de valores de intensidade, quais sejam: Descontinuidade e a Similaridade.

A descontinuidade, se baseia na divisão de uma imagem com base nas mudanças bruscas de intensidade, como as bordas. Já a similaridade, se baseia na divisão de uma

Figura 4 – Passos fundamentais em PDI.



Fonte: Processamento Digital de Imagens 3ª Edição ([GONZALEZ; WOORS, 2010](#))

imagem em regiões que sejam semelhantes de acordo com um conjunto de critérios predefinidos.

([SANNIHIT; PURI; SINGH, 2021](#)), aponta para as principais técnicas de segmentação de imagem, que são as baseadas em: Bordas; Fuzzy; Equações Diferenciais Parciais; Limiar (*threshold*); Região; Segmentação baseada em Cluster; Segmentação baseada em Redes Neurais Convolucionais (RNC).

Assim, ([GONZALEZ; WOORS, 2010](#)) oferecem um resumo matemático para a compreensão dos fundamentos da segmentação:

- Considere que  $R$  é toda a região espacial ocupada por uma imagem;  $Q(R_k)$  é uma propriedade lógica definida sobre os pontos no conjunto  $R_k$ ;  $\theta$  é o conjunto nulo; duas regiões  $R_i$  e  $R_j$  são ditas adjacentes se a sua união forma um conjunto conexo;
- Segmentação de imagem é um processo que particiona  $R$  em  $n$  sub-regiões,  $R_1, R_2, \dots, R_n$ , de tal forma que:

1.  $\bigcup_{i=1}^n R_i = R$

2.  $R_i$  é um conjunto conectado,  $i = 1, 2, \dots, n$

3.  $R_i \cap R_j = \theta$  para todo  $i$  e  $j$ ,  $i \neq j$
4.  $Q(R_i) = TRUE$  para  $i = 1, 2, \dots, n$
5.  $Q(R_i \cap R_j) = FALSE$  para quaisquer regiões adjacentes  $R_i$  e  $R_j$

Sendo mais específico em relação aos itens “4” e “5”, “4” tem relação com as propriedades que devem ser cumpridas pelos pixels em uma região segmentada; enquanto “5” indica que duas regiões adjacentes,  $R_i$  e  $R_j$  devem ser diferentes no sentido da propriedade  $Q$ .

Logo, o problema fundamental da segmentação está relacionado a divisão de uma imagem em regiões que satisfaçam as condições anteriores.

Os métodos de segmentação são predominantemente classificados em três categorias: Segmentação Semântica, Segmentação por Instância e Segmentação Panóptica.

A segmentação semântica analisa cada pixel da imagem e atribui uma única classe de rótulo com base na textura que representa, ou seja, ele segmenta o objeto por classe.

Já a segmentação por instância tem a capacidade de detectar cada objeto ou instância de classe presentes na imagem e atribuir a cada um uma máscara ou caixa delimitadora distinta, acompanhada de um identificador único.

A segmentação panóptica adota uma abordagem unificada, atribuindo a cada pixel da cena um rótulo semântico (graças a segmentação semântica) e um identificador de instância único (devido à segmentação por instância).

### 2.2.2 Sob a ótica da disciplina de inteligência artificial

Conforme (RUSSELL; NORVIG, 2009), o principal tema unificador é a ideia de agente inteligente. Assim, Inteligência Artificial (IA) é definido como “o estudo de agentes que recebem percepções do ambiente e executam ações”. Deste modo, o campo da IA não tenta apenas compreender entidades inteligentes, mas também construí-las.

Um agente é definido como algo que age. Assim, espera-se que um agente computacional opere sob controle autônomo, perceba seu ambiente, persista por um período de tempo prolongado, adapte-se a mudanças e seja capaz de criar e perseguir metas. A IA busca pelo agente racional que é definido como “aquele que age para alcançar o melhor resultado ou, quando há incerteza, o melhor resultado esperado”.

As disciplinas que contribuíram com as ideias, pontos de vista e técnica para a IA, são: (1) A filosofia, que tornou a IA concebível, pois considera a ideia de que a mente é, em alguns aspectos, semelhante a uma máquina; (2) A matemática, que definiu a base para a compreensão da computação e do raciocínio sobre algoritmos; (3) A economia, que contribuiu na formalização do problema de tomar decisões que maximizam o resultado

esperado para o tomador de decisões; (4) A neurociência, que descobriu alguns fatos sobre como a mente trabalha e a forma como ela se assemelha e se diferencia dos computadores; (5) A psicologia, que adotou a ideia que os seres vivos podem ser considerados máquinas de processamento de informações; (6) A Engenharia da computação, que fornece máquinas cada vez mais poderosas.

Algumas aplicações da IA envolvem: (1) Veículos robóticos; (2) Reconhecimento de voz; (3) Planejamento autônomo e escalonamento; (4) Jogos; (5) Combate a spam; (6) Planejamento logístico; (7) Robótica; (8) Tradução automática. Outro fato importante é que, em termos matemáticos, o comportamento do agente é descrito pela função do agente, ou seja, as ações programadas dependem das percepções do agente.

É possível inferir que o campo da IA é bastante amplo e seu estudo depende dos objetivos. Assim, o interesse está no estudo de um agente de software ou softbot que recebe um conjunto de imagens como entradas sensoriais e atua sobre o ambiente exibindo o resultado das operações de segmentação, comparação de máscaras reais e previstas, e exibição dos resultados conforme métricas a serem estabelecidas.

Dentre os conteúdos cobertos pela IA, a pesquisa realizada apontou para o ramo de aprendizagem, conhecido como *Machine Learning*.

### 2.2.2.1 *Machine Learning*

Simplifica-se a ideia de *Machine Learning* como sistemas que modificam seu comportamento, de forma autônoma, de acordo com suas próprias experiências. Tais experiências são adquiridas através de treinamento.

Dentre as várias definições possíveis, buscou-se uma mais ampla sobre aprendizagem, aplicado no contexto da computação, conforme segue: Diz-se que um programa de computador aprende com a experiência  $E$  em relação a alguma classe de tarefas  $T$  e com uma medida de desempenho  $P$ , se o seu desempenho nas tarefas em  $T$ , conforme medido por  $P$ , melhora com a experiência  $E$  (MITCHELL, 1997).

A modificação do comportamento ocorre segundo as regras lógicas estabelecidas que podem tanto melhorar o desempenho de uma tarefa quanto tomar a decisão mais apropriada para o contexto. Essas regras são geradas com base no reconhecimento de padrões dentro dos dados analisados

Qual a utilidade de aprendizagem em um sistema? Se o projeto do sistema pode ser melhorado, por que os projetistas não apenas programam essa melhoria? (RUSSELL; NORVIG, 2009) oferecem três razões para tais questionamentos: (1) “Os projetistas não podem antecipar todas as situações possíveis em que o agente possa se encontrar”; (2) “Os projetistas não podem antecipar todas as mudanças ao longo do tempo” e; (3) “Os programadores humanos não têm ideia de como programar uma solução por si só”.

Dentre as abordagens estabelecidas em *Machine Learning*, concentra-se no ramo que obteve sucesso considerável nas últimas décadas, o *Deep Learning*.

#### 2.2.2.2 *Deep Learning*

Segundo (GOODFELLOW; BENGIO; COURVILLE, 2016), consideram como *Deep Learning* o estudo de modelos que envolvem uma quantidade maior de composição de funções ou conceitos aprendidos comparado ao *Machine Learning* tradicional.

Assim, a *Deep Learning* é um tipo particular de *Machine Learning*, pois é uma técnica que permite sistemas de computadores se aprimorarem com experiência e dados. Ao aprender a representar o mundo como uma hierarquia aninhada de conceitos, alcança grande poder e flexibilidade. Cada conceito é definido em relação a conceitos mais simples enquanto as representações mais abstratas são computadas em termos de conceitos menos abstratos.

Conforme os autores, os algoritmos de *Deep Learning* são baseados no algoritmo de otimização conhecido como descida de gradiente estocástico. Assim, o descreve como a combinação de vários algoritmos tais como: um algoritmo de otimização; um sobre a função de custo; um modelo e; um conjunto de dados; para construir um algoritmo de *Machine Learning*.

Para entender os detalhes da *Deep Learning* é necessário conhecer sua estrutura. Tal estrutura é formada pelo conceito de redes neurais profundas. Este conceito tem uma forma mais simples, conhecida como rede neural.

Assim, dedica-se a próxima parte ao estudo das redes neurais e seus componentes.

#### 2.2.2.3 Redes Neurais

Para (AGGARWAL, 2018) redes neurais artificiais são técnicas populares de *Machine Learning* que simulam o mecanismo de aprendizagem dos organismos biológicos. Assim, a rede tenta simular o sistema nervoso humano que contém neurônios, estes conectados a outros neurônios através do axônios e dendritos que, por sua vez, se comunicam através de sinapses. Estímulos externos podem modificar a força das conexões sinápticas e essa mudança é como um organismo vivo aprende.

(RUSSELL; NORVIG, 2009) resumem a explicação da hipótese da neurociência: “a atividade mental consiste basicamente na atividade eletroquímica em redes de células cerebrais chamadas neurônios.” Assim, com inspiração nessa hipótese, direcionou-se esforços para a criação de redes neurais artificiais.

Para (NIELSEN, 2015), uma rede neural artificial aprende usando exemplos para automaticamente inferir regras. Quanto maior o número de exemplos melhor o seu aprendizado e a sua precisão.

As ideias fundamentais sobre redes neurais, incluem dois tipos de neurônios artificiais (o perceptron e o neurônio sigmoid) e o seu algoritmo padrão de aprendizagem, conhecido como descida de gradiente estocástico.

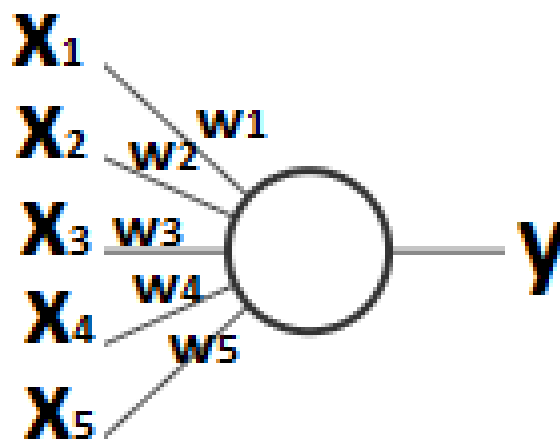
### 2.2.2.3.1 Perceptrons

Explicar o que é uma rede neural, implica entender sobre um tipo de neurônio artificial denominado *Perceptron*. O cientista responsável por desenvolvê-lo, entre os anos de 1950 e 1960, chamado Frank Rosenblatt, inspirou-se nos primeiros trabalhos de Warren McCulloch e Walter Pitts. Apesar de existirem modelos mais completos, o *Perceptron* oferece as bases para a compreensão desses modelos.

A quantidade de entradas em um *Perceptron* varia de acordo com o projeto, sendo generalizado como  $x_1, x_2, \dots, x_n$ . Em contrapartida, possui apenas uma saída.

No exemplo da Fig. 5, o *Perceptron* possui cinco entradas,  $x_1, x_2, x_3, x_4, x_5$  e uma saída  $y$ , conforme modelo abaixo.

Figura 5 – *Perceptron*.



Fonte: De autoria própria

Para computar a saída, Rosenblatt propôs a introdução dos pesos (*weight*),  $w_1, w_2, \dots, w_n$ , expressos em números reais, indicando a importância de cada entrada.

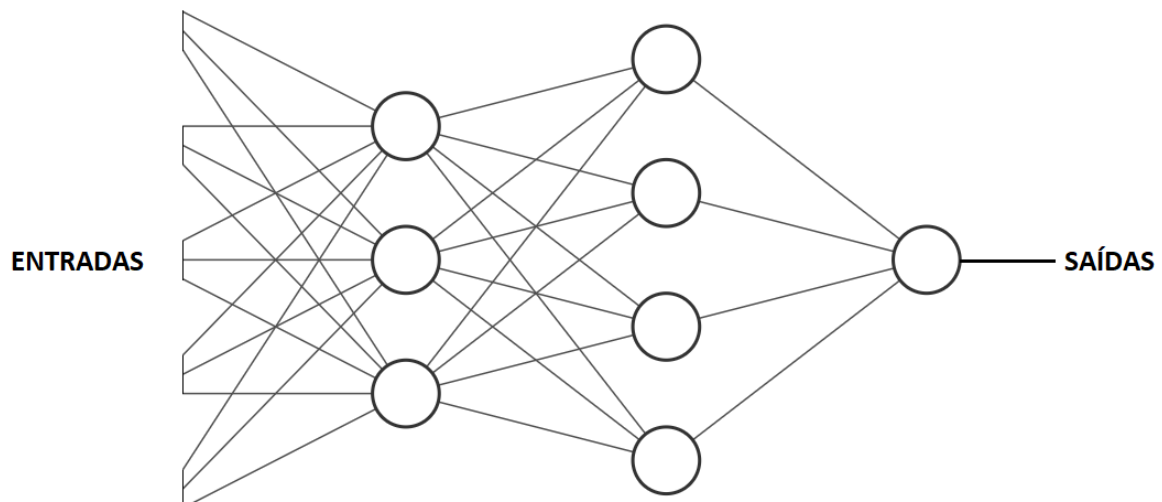
Tanto as entradas quanto a saída são expressas por um número binário, 0 ou 1. O resultado do somatório  $\sum w_j x_j$  é comparado com um valor de limiar (*threshold*), sendo este um parâmetro do neurônio dado em números reais. Podemos resumir-lo em termos algébricos:

$$y = \begin{cases} 0 & \text{se } \sum w_j x_j \leq \text{threshold} \\ 1 & \text{se } \sum w_j x_j > \text{threshold} \end{cases} \quad (2.1)$$

Assim, variando os valores dos pesos e limiares, obtêm-se modelos diferentes e por consequência altera o modo de fazer decisões. Porém, um neurônio artificial isolado está distante de executar uma decisão complexa satisfatória. O exemplo apresentado serve para ilustrar como um *Perceptron* pode pesar diferentes tipos de evidências para tomar decisões.

É plausível imaginar que construir uma rede de *Perceptrons* eleva o nível das decisões.

Figura 6 – Rede de *Perceptrons*.



Fonte: Imagem adaptada de NN-SVG ([LENAIL, 2023](#))

Analisando a Fig. 6 é notável a existência de oito *Perceptrons*, sendo estes dispostos em três colunas. Na literatura, cada coluna é entendida como uma camada, sendo enumeradas da posição de entrada para a posição de saída. De forma intuitiva, a primeira camada do exemplo realiza três decisões muito simples que se baseiam nos pesos de suas entradas. Na segunda camada, os *Perceptrons* recebem os resultados da primeira camada adicionado aos pesos da segunda camada. Esta ação, torna as decisões da segunda camada mais complexas e mais abstratas em comparação com a primeira. E seguindo a ideia desenvolvida, a terceira é capaz de realizar decisões ainda mais complexas.

Considerando o exemplo dado, é possível deduzir que quanto maior o número de camadas melhor o nível das decisões.

Sobre a representação visual da Fig. 6, considere que os *Perceptrons* da primeira e da segunda camada possuem apenas uma saída. Essa é uma forma simples de representar que a sua saída é utilizada por todas as entradas da camada subsequente.

Substituindo o somatório  $\sum w_j x_j$  da inequação 2.1 pelo produto vetorial  $w \cdot x$ , sendo  $w$  e  $x$  vetores de componentes dos pesos e das entradas, respectivamente, obtém-se o resultado  $w \cdot x = \sum w_j x_j$ . Movendo o valor do limiar para o outro lado da inequação, obtém-se um limiar negativo. Transformando o valor do limiar em outra variável, denominado viés (*bias*), obtém-se,  $b \equiv -threshold$ . Assim, após as devidas modificações, obtém-se a regra a seguir:

$$y = \begin{cases} 0 & \text{se } \sum w \cdot x + b \leq 0 \\ 1 & \text{se } \sum w \cdot x + b > 0 \end{cases} \quad (2.2)$$

Conforme (NIELSEN, 2015), o viés mede a facilidade de um *Perceptron* produzir na saída o valor 1. Quanto maior o viés, maiores são as probabilidades do *Perceptron* ativar, ou seja, produzir o valor 1. Quanto menor o viés, maiores são as probabilidades do *Perceptron* desativar, ou seja, produzir o valor 0.

### 2.2.2.3.2 Neurônios sigmoid

Neurônios sigmoid são similares aos *Perceptrons*, mas modificados para que pequenas mudanças nos pesos e vieses causem apenas pequenas mudanças em suas saídas. Isso é um fato importante que permite ao neurônio sigmoid aprender.

A estrutura do neurônio sigmoid é apresentada na Fig. 7. A quantidade de entradas, assim como o *Perceptron*, varia de acordo com o projeto, sendo generalizado como  $x_1, x_2, \dots, x_n$ . Entretanto, seus valores não são binários, podendo variar em qualquer faixa entre 0 e 1. Tanto seus pesos quanto seus vieses operam como os do *Perceptron*. As saídas não são apenas valores binários, mas através da função  $\sigma(wx + b)$ , onde  $\sigma$  é chamado função sigmoid, obtém-se os valores da saída:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.3)$$

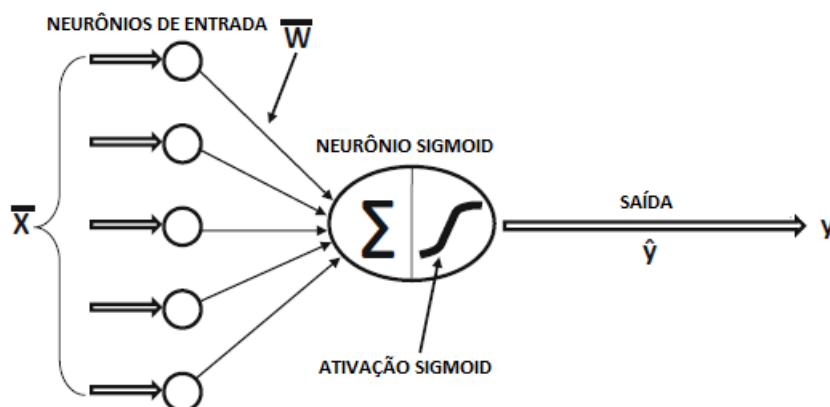
Substituindo  $z$  por  $wx + b$ , obtém-se a expressão:

$$\sigma(w \cdot x + b) = \frac{1}{1 + \exp(-\sum_j w_j x_j - b)} \quad (2.4)$$

A função 2.3 e 2.4 possui uma forma semelhante ao apresentado na Fig. 8. A forma exata da função sigmoid não é tão importante, mas sim o formato da função quando plotada.

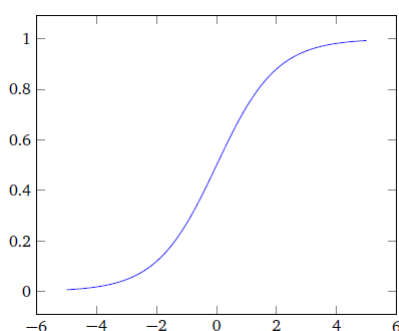


Figura 7 – Neurônio Sigmoid.



Fonte: Imagem adaptada de *Neural Networks and Deep Learning* (AGGARWAL, 2018)

Figura 8 – Função Sigmoid



Fonte: *Neural Networks and Deep Learning* (NIELSEN, 2015)

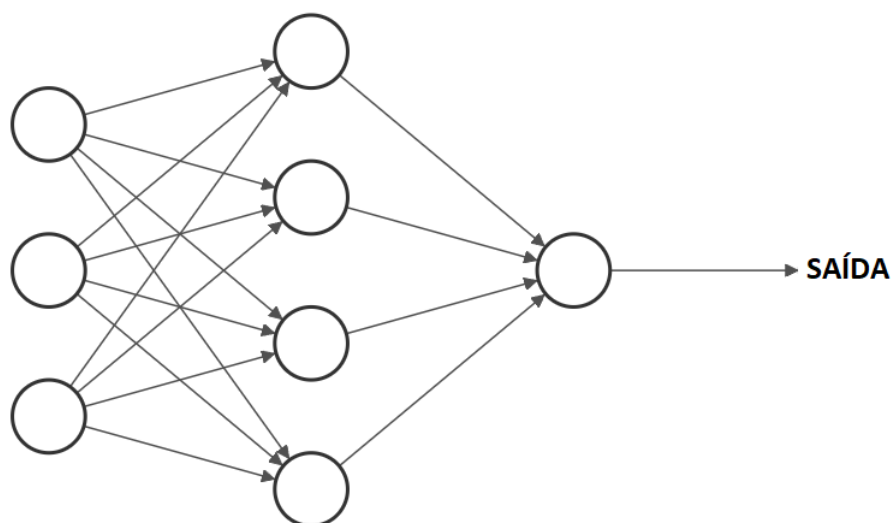
Assim, considerando que a saída são valores entre 0 e 1, é possível interpretá-las como valores probabilísticos de ser ou não ser o objeto definido para uma determinada análise, inserindo graus de certeza e incerteza.

### 2.2.2.3.3 A arquitetura das redes neurais

Após conhecer os neurônios artificiais é possível entender melhor sobre as redes neurais.

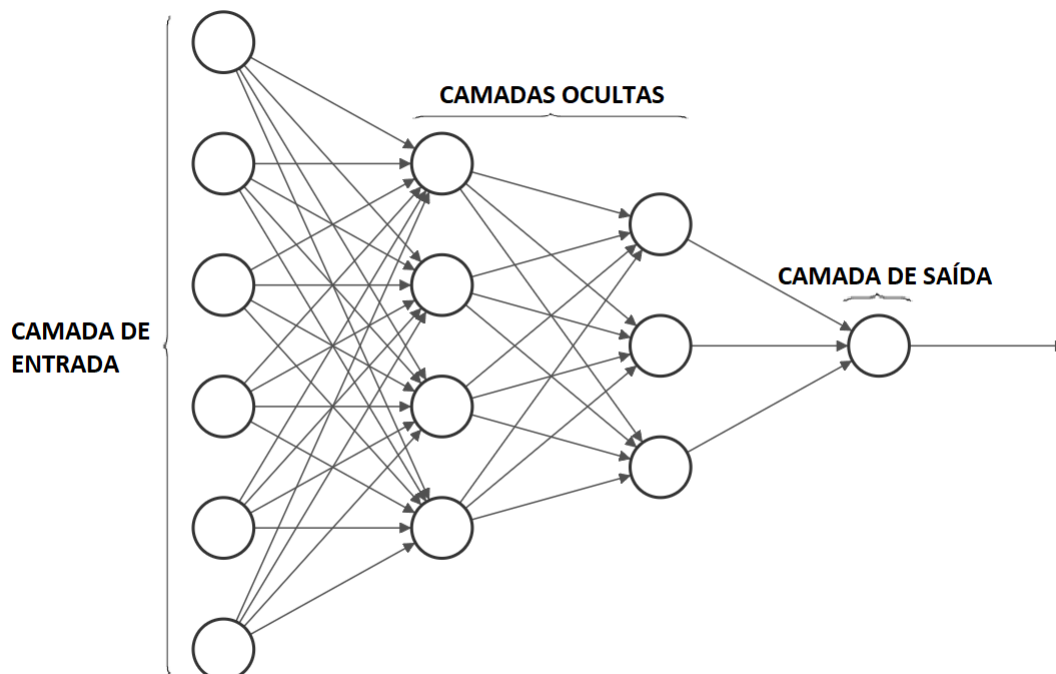
Como mencionado anteriormente, na Fig. 9 existem oito *Perceptrons* dispostos em três camadas. A camada na extremidade esquerda é denominada de camada de entrada e a camada na extremidade direita é denominada de camada de saída. Qualquer camada entre as duas camadas anteriores, é chamada de camada escondida. Logo, a rede neural em questão, possui apenas uma camada escondida. Esse exemplo, ilustra uma rede neural rasa e, considerando a quantidade unitária de camada escondida, não entra no conceito de *Deep Learning*.

Figura 9 – Arquitetura de rede neural



Fonte: Imagem adaptada de NN-SVG ([LENAIL, 2023](#))

Figura 10 – Arquitetura de rede neural multicamadas



Fonte: Imagem adaptada de NN-SVG ([LENAIL, 2023](#))

A Fig. 10 trata-se de uma rede com múltiplas camadas. Algumas vezes são chamadas de multilayer *Perceptrons* (*Perceptrons* multicamadas) ou MLP, mesmo sendo composto

por neurônios sigmoid e não por *Perceptrons*.

À medida que aumentamos a quantidade de camadas escondidas dentro da rede neural projetada, aumentamos a sua profundidade.

Projetar tanto a entrada quanto a saída é frequentemente uma tarefa simples. Como exemplo, uma imagem em escala de cinza que possui uma quantidade de  $32 \times 32$  pixels deve possuir  $1.024 = 32 \times 32$  neurônios na camada de entrada, os respectivos valores de intensidade dos pixels devem ser normalizados para valores entre 0 a 1. A quantidade de neurônios na camada de saída depende de quantas características se deseja extrair.

O projeto da camada escondida depende do comportamento desejado para a rede. Uma série de pesquisadores desenvolvem muitas heurísticas para essa camada, que ajudam as pessoas a conseguir tal comportamento.

Assim, observa-se que o conceito de *Deep Learning* está ligada diretamente ao conceito de redes neurais profundas, ou seja, redes com muitas camadas.

#### 2.2.2.3.4 Algoritmos de retropropagação (*backpropagation*) e descida do gradiente

(COPPIN, 2010) discorre sobre o *backpropagation*. A principal diferença entre uma rede multicamada e um *Perceptron* simples é que cada neurônio possui pesos associados às suas entradas e, por consequência, há muito mais pesos a serem ajustados quando um erro é cometido para um lote de dados de treinamento. Nesse contexto, o método *backpropagation* vem para atribuir responsabilidade aos diferentes pesos.

Redes multicamada com *backpropagation* geralmente usam a função sigmóide  $\sigma(x)$ , pois a função é diferenciável, conforme descrito a seguir:

$$\frac{d\sigma(x)}{dx} = \sigma(x) \cdot (1 - \sigma(x)) \quad (2.5)$$

O algoritmo de *backpropagation* inicia atribuindo valores aleatórios aos pesos na rede. A cada iteração envolve, de início, a alimentação de dados por meio da rede, desde as entradas até as saídas. A próxima fase envolve a alimentação de erros de volta, das saídas às entradas. À medida que os valores de erro retornam através da rede, ocorre alteração nos pesos dos neurônios ao longo do caminho. O algoritmo mantém esse comportamento até que os valores de erro sejam suficientemente pequenos.

Como exemplo, considere uma rede de três camadas, onde  $i$  representa os neurônios da camada de entrada,  $j$  representa os neurônios da camada oculta e  $k$  representa os neurônios da camada de saída.

A função usada para gerar o valor da saída de um nó  $j$  na rede pode ser expressa como:

$$X_j = \sum_{i=1}^n x_i \cdot w_{ij} - b_j \quad (2.6)$$

$$Y_j = \frac{1}{1 + e^{-X_j}} \quad (2.7)$$

sendo:

- $n$  o número de entradas para o neurônio  $j$ ;
- $w_{ij}$  o peso da conexão entre cada nó  $i$  e cada neurônio  $j$ ;
- $b_j$  é o viés em uso para o neurônio  $j$ , estabelecido entre o valor aleatório de 0 a 1;
- $x_i$  é o valor de entrada para o neurônio de entrada  $i$ ;
- $Y_j$  é o valor de saída produzido pelo neurônio  $j$ .

Após os neurônios receberem as entradas para computarem as saídas, um gradiente de erro é calculado para cada neurônio  $k$  na camada de saída.

O sinal de erro para  $k$  é definido pela diferença entre o valor real e o valor predito para aquele neurônio:

$$e_k = r_k - y_k \quad (2.8)$$

sendo:

- $r_k$  o valor real para o neurônio  $k$ ;
- $y_k$  o valor predito desta iteração.

O gradiente de erro para o neurônio de saída  $k$  é definido como o sinal de erro para este neurônio, multiplicado pela derivada da função de ativação:

$$\delta_k = \frac{\partial y_k}{\partial x_k} \cdot e_k \quad (2.9)$$

onde  $x_k$  é a soma ponderada dos valores de entrada do neurônio  $k$ .

Considerando que  $y$  é definido como uma função sigmóide em  $x$ , utiliza-se a fórmula dada anteriormente para a derivada da função sigmóide para obter a seguinte fórmula para o gradiente de erro:

$$\delta_k = y_k \cdot (1 - y_k) \cdot e_k \quad (2.10)$$

De modo semelhante, calcula-se um gradiente de erro para cada neurônio  $j$  na camada oculta, conforme segue:

$$\delta_j = y_k \cdot (1 - y_k) \sum_{k=1}^n x_j \cdot w_{jk} - \delta_k \quad (2.11)$$

sendo  $n$  o número de neurônios na camada de saída.

O último passo, consiste na atualização de cada peso na rede,  $w_{ij}$  ou  $w_{jk}$ , conforme a fórmula a seguir:

$$w_{ij} \leftarrow w_{ij} + \alpha \cdot x_i \cdot \delta_j \quad (2.12)$$

$$w_{jk} \leftarrow w_{jk} + \alpha \cdot y_j \cdot \delta_k \quad (2.13)$$

sendo:

- $x_i$  o valor de entrada para o nó  $i$ ;
- $\alpha$  é a taxa de aprendizado.

O método apresentado é conhecido por descida do gradiente. O seu comportamento envolve descer a superfície que representa a função de erro pelo caminho mais íngreme, com o objetivo de alcançar o mínimo dentro do espaço de erro. Este comportamento altera o conjunto de pesos da rede, fornecendo um melhor desempenho para a rede.

([RUSSELL; NORVIG, 2009](#)) oferece um resumo do processo de *backpropagation*:

- Calcule valores  $\delta$  para os neurônios de saída usando o erro observado.
- A partir da camada de saída, repita o seguinte para cada camada da rede até que a primeira camada oculta seja alcançada:
  - Propague o valores  $\delta$  de volta à camada anterior.
  - Atualize os pesos entre as duas camadas.

#### 2.2.2.4 Redes Neurais Convolucionais

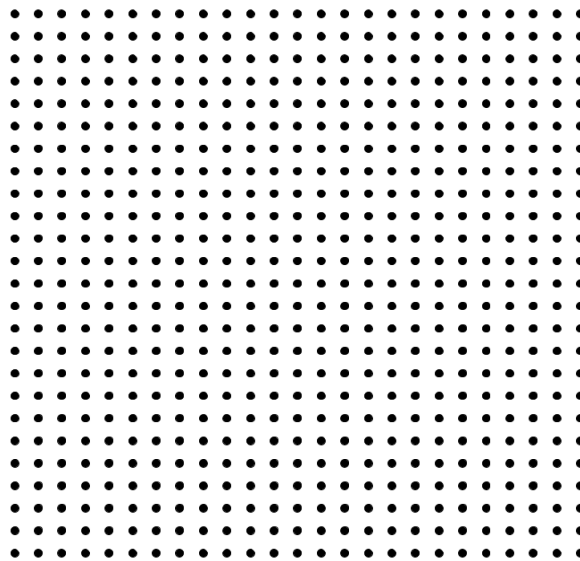
De acordo com ([NIELSEN, 2015](#)), as redes neurais convolucionais apresentam uma arquitetura especializada e bem adaptada para a classificação de imagens. Sendo assim, redes convolucionais profundas ou suas variantes são amplamente empregadas em diversas aplicações para o reconhecimento de imagens.

Estas redes, utilizam três ideias básicas: campos receptivos locais, pesos e vieses compartilhados e agrupamento (*pooling*).

### Campos receptivos locais:

Os exemplos anteriores representaram a camada de entrada com neurônios enfileirados em uma coluna. Em redes convolucionais, simplifica pensar que os neurônios de entrada estão dispostos em uma matriz de tamanho  $n \times m$  pixels de intensidade. Observe a Fig. 11 que representa uma matriz de tamanho  $25 \times 25$ :

Figura 11 – Neurônios de Entrada



Fonte: De autoria própria

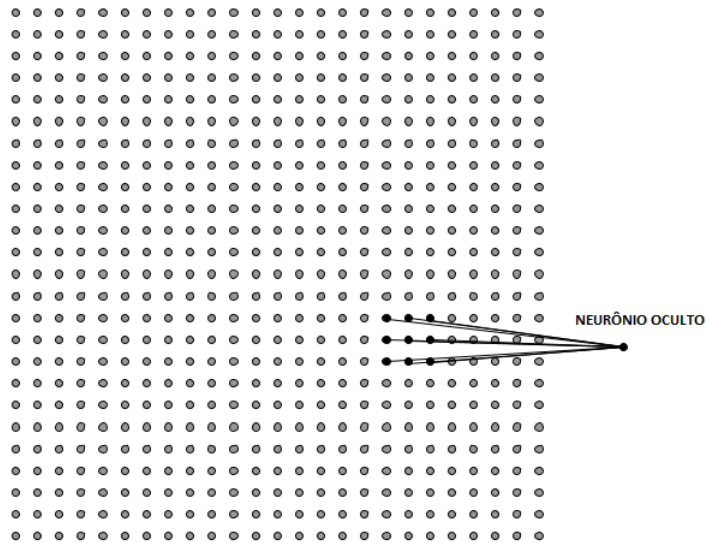
Em redes neurais convencionais, cada neurônio na camada de entrada se conecta a todos os demais neurônios da camada seguinte. Em contrapartida, as redes neurais convolucionais agrupam neurônios próximos ao neurônio central no qual se cria uma pequena região localizada no formato de uma matriz  $n \times n$ , normalmente muito menor que a imagem de entrada, que se conecta ao neurônio da camada seguinte.

Conforme Fig. 12, escolheu-se uma região de tamanho  $3 \times 3$ , correspondente a 25 pixels de entrada. É importante perceber que o número de neurônios de entrada ligados à próxima camada é menor que em uma rede neural convencional. Por fim, esta pequena região é conhecida como campo receptivo local para o neurônio escondido.

Cada conexão aprende um peso e os neurônios escondidos aprendem um viés geral. Assim, um neurônio escondido particular está aprendendo a analisar um campo receptivo local específico.

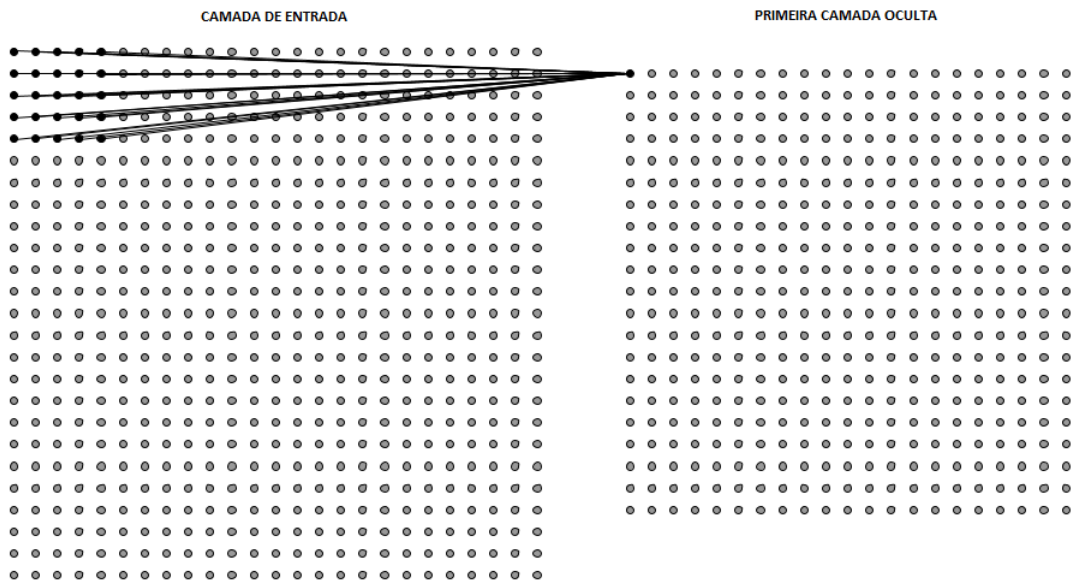
Semelhante a operação de convolução, deslocamos a matriz ao longo da imagem de entrada. A cada deslocamento implica a existência de um neurônio escondido na próxima camada.

Figura 12 – Campo Receptivo Local



Fonte: De autoria própria

Figura 13 – Ligação entre a camada de entrada e a camada oculta utilizando um campo receptivo local 5x5



Fonte: De autoria própria

No exemplo da Fig. 13, uma imagem de entrada de 25x25 e um campo receptivo local 5x5 possui uma camada de neurônios escondidos de 21x21. Isso ocorre porque os neurônios da borda não possuem neurônios próximos o suficiente para formar uma matriz 5x5, sendo, neste caso, ignorados.

Vale ressaltar o número de passos (*stride*) da operação acima realizada. No exemplo

dado, demonstra-se passos de tamanho 1, ou seja, a matriz desliza sempre para o neurônio adjacente. Em casos onde os passos são configurados em ordem maiores, a matriz desliza uma quantidade de neurônios equivalente ao número de passos definido.

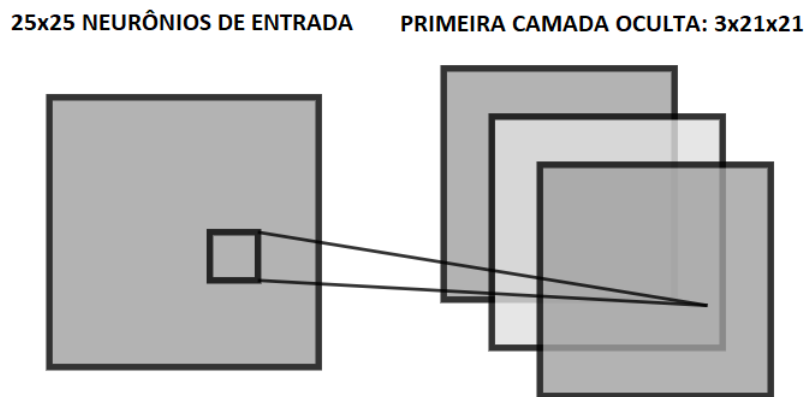
### **Pesos e vieses compartilhados:**

Considerando o exemplo anterior, os pesos e vieses são os mesmos para cada um dos 21x21 neurônios escondidos presentes na camada. A operação de convolução é parâmetro de entrada para uma dada função de ativação escolhida. Devido a esta operação que a rede possui o nome convolucional em sua denominação.

Conforme à característica apresentada, todos os neurônios da primeira camada escondida detectam exatamente os mesmos recursos e, por vezes, denomina-se o mapeamento da camada de entrada para a camada escondida como mapa de recursos (*feature map*). Também, os pesos e vieses compartilhados, frequentemente são definidos como *kernel* ou filtros.

A estrutura da rede para reconhecimento de imagem precisa de vários mapas de recursos e uma camada convolucional completa consiste de uma severa quantidade desses mapas. A cada mapa de recursos é possível detectar um tipo diferente de característica da imagem de entrada.

Figura 14 – Divisão em mapas de recursos



Fonte: Imagem adaptada de NN-SVG ([LENAIL, 2023](#))

A grande vantagem do conceito de pesos e vieses compartilhados é a alta redução do número de parâmetros necessários para a rede convolucional. Isto implica em um rápido treinamento para os modelos convolucionais ajudando a construir redes mais profundas.

### **Camadas de agrupamento:**

A camada de agrupamento normalmente é usada imediatamente após a camada convolucional. Sua finalidade é simplificar informações da saída da camada convolucional.

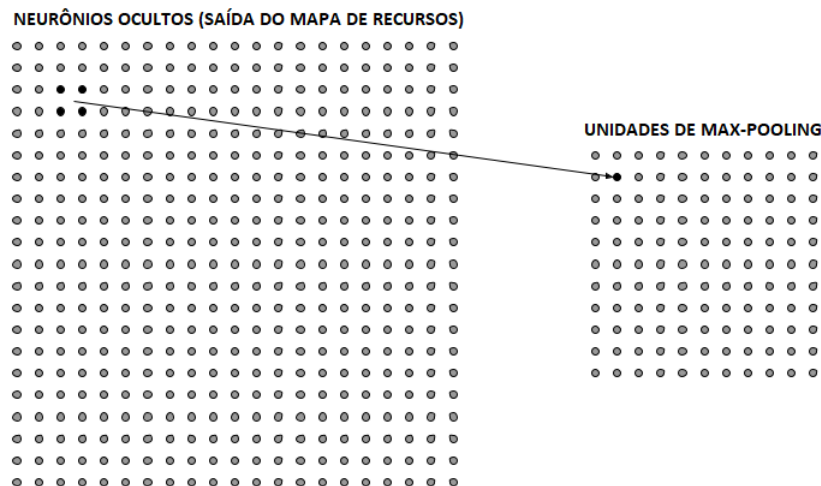


Semelhante aos Wavelets, mais precisamente a técnica de interpolação, referente a PDI, que consiste em diminuir ou acrescentar informações na imagem manipulada, a partir das características existentes na mesma, aumentando ou diminuindo sua resolução, a camada de agrupamento condensa as informações dos mapas de recursos.

Ilustra-se a ideia com o seguinte exemplo. Por hipótese, após a imagem de entrada ser processada através da camada convolucional, extrai-se determinada quantidade de mapas de recursos. A camada de agrupamento extrai uma região  $n \times n$ , normalmente  $2 \times 2$ , do mapa de recursos e resume a matriz para um único valor que será encaminhada para a próxima camada.

Um procedimento amplamente utilizado na camada de agrupamento é conhecido como *max-pooling*. Considerando uma região  $2 \times 2$ , essa operação extrai o maior valor existente dentro da matriz e encaminha para a próxima camada, conforme ilustrado abaixo:

Figura 15 – Extraindo o valor mais significativo na operação max-pooling



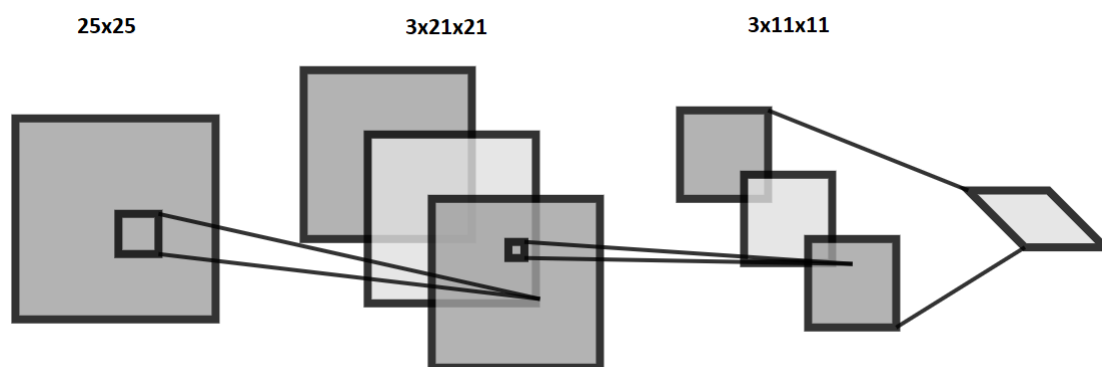
Fonte: De autoria própria

Considerando um mapa de recursos  $21 \times 21$ , após a operação utilizando *max-pooling*, obtém-se uma matriz de neurônios  $11 \times 11$ .

Em posse do conhecimento sobre as ideias que fundamentam a construção da Rede Neural Convolucional, é possível imaginar a forma da rede, ilustrado na Fig. 16:

A rede apresentada inicia com  $25 \times 25$  neurônios de entrada que são utilizados para codificar os pixels de intensidade. Após operações na camada convolucional utilizando campos receptivos locais de tamanho  $5 \times 5$ , são gerados três mapas de recursos de tamanho  $21 \times 21$ . O próximo passo envolve a camada de *max-pooling*, aplicado em regiões  $2 \times 2$  em cada um dos três mapas de recursos. O resultado é uma camada de  $3 \times 11 \times 11$  neurônios da camada oculta. A última camada apresentada, opera exatamente igual as redes neurais

Figura 16 – Rede Neural Convolutiva



Fonte: Imagem adaptada de NN-SVG ([LENAIL, 2023](#))

clássicas apresentadas até então.

#### 2.2.2.5 U-Net

([RONNEBERGER; FISCHER; BROX, 2015](#)) desenvolveram a U-Net, um modelo de rede neural convolutiva, afirmando se tratar de uma Rede Completamente Convolutiva.

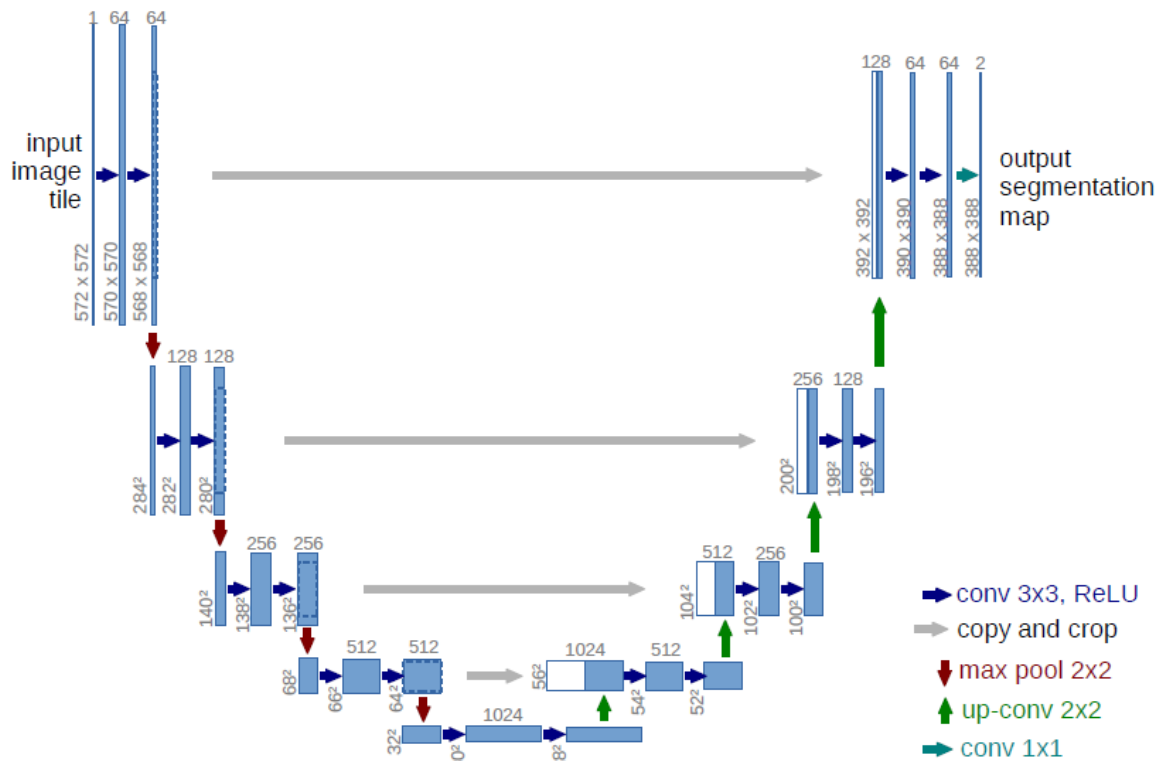
A arquitetura da rede é ilustrada na Fig. 17 e consiste em dois componentes principais, um caminho de contração (lado esquerdo) e um caminho de expansão (lado direito). O caminho de contração (*Encoder*) segue uma estrutura convencional de redes convolucionais, com convoluções  $3 \times 3$  seguidas de ativação *ReLU* e *max-pooling*  $2 \times 2$ , dobrando o número de mapas de recursos em cada redução de resolução (*downsampling*).

Após a fase de contração, o caminho de expansão (*Decoder*) inicia-se com operações de aumento de resolução (*upsampling*) nos mapas de recurso, seguido por uma convolução  $2 \times 2$  que reduz pela metade o número de mapas de recurso. Cada etapa inclui a concatenação com o mapa de recurso correspondente do caminho de contração, seguida por duas convoluções  $3 \times 3$ , cada uma com ativação *ReLU*. O recorte é essencial para preservar os pixels de borda perdidos durante a convolução. Na última camada, uma convolução  $1 \times 1$  mapeia cada vetor de recurso de 64 componentes para o número desejado de classes. A rede possui um total de 23 camadas convolucionais.

Conforme representado na Fig. 17, observa-se que o caminho de expansão é simétrico ao caminho de contração, resultando em uma arquitetura em forma da letra “U”. Essa simetria inspirou os autores a nomearem o modelo como U-Net.

Em busca de maior formalidade, é primordial compreender que o modelo U-Net é comumente empregado em tarefas de segmentação semântica. Seu objetivo é classificar cada pixel em uma imagem de entrada em diferentes classes e categorias. A formulação do problema associado à U-Net pode ser descrita da seguinte maneira:

Figura 17 – O modelo U-Net



Fonte: *U-Net: Convolutional Networks for Biomedical Image Segmentation*.  
(RONNEBERGER; FISCHER; BROX, 2015)

**Problema:** Dado uma imagem de entrada  $X$ , busca-se segmentar a imagem em  $K$  classes, onde  $K$  é o número de classes. A saída deve ser uma máscara de segmentação  $Y$  com o mesmo tamanho da imagem de entrada, em que cada pixel é atribuído uma classe de rótulos.

**Entrada:**  $X$  é a imagem de entrada com dimensões de  $ALC$ , sendo  $A$  a altura,  $L$  a largura e  $C$  o número de canais.

**Saída:**  $Y$  é a máscara de segmentação com dimensões de  $ALK$ , sendo  $K$  o novo conjunto de canais. Cada canal representa a probabilidade de predição da classe correspondente para cada pixel.

**Modelo:** O modelo U-Net consiste em dois caminhos principais: o caminho de contração (*encoder*) e o caminho de expansão (*decoder*).

- **Dupla convolução seguida da função de ativação (*ReLU*):**

- Consiste em repetir duas vezes a operação de convolução seguida da função de ativação (*ReLU*).

$$C_j = Conv(C_{j-1}, f_j)$$

$$R_j = ReLU(C_j)$$

$$C_{j+1} = Conv(R_j, f_{j+1})$$

$$R_{j+1} = ReLU(C_{j+1})$$

– Assim,  $Conv2x(X) = R_{j+1}$

- **Caminho de contração (*Encoder*):**

– A primeira etapa, consiste na operação de convolução seguida da função de ativação (*ReLU*), repetida duas vezes. Em seguida, o resultado é submetido à operação de *max-pooling*.

– O *Encoder* aprende a captar recursos hierárquicos e reduzir as dimensões espaciais.

– A cada camada, o número de filtros ou aumenta ou permanece constante.

$$C_i = Conv2x(C_{i-1})$$

$$P_i = MaxPooling(C_i)$$

- **Gargalo (*Bottleneck*):**

– Consiste em executar  $Conv2x(Z)$

– Uma camada central que captura características mais abstratas.

$$B = Conv2x(P_n)$$

- **Caminho de expansão (*Decoder*):**

– Consiste em aumentar a resolução (através da operação de convolução transposta) seguido pela concatenação com seus mapas de recursos correspondentes (mesma dimensão) advindos do caminho de contração.

– O *Decoder* aprende a refinar detalhes espaciais.

$$U_i = UpConv(B, C_{n-i}, f_{n+i+1})$$

$$C_{n+i+1} = Conv2x(U_i)$$

- **Saída:**

– A última camada faz a predição da máscara de segmentação para cada classe.

$$Y = Conv(C_{n+i+1}, K)$$

### 2.2.2.6 Funções de Perda e Métricas de Desempenho

Com base nos estudos de (TERVEN et al., 2023), um dos componentes críticos para o sucesso na *Deep Learning* é a escolha cuidadosa da função de perda (*Loss Function*) e da métrica de desempenho (*Performance Metric*) utilizadas para treinar e avaliar o modelo.

Apesar de possuírem semelhanças ao avaliar o desempenho de um modelo de *Deep Learning*, as funções de perda e as métricas desempenham papéis distintos. A função de perda é empregada durante o treinamento para otimizar os parâmetros do modelo, medindo a diferença entre a saída predita e a saída esperada, com o objetivo de minimizar essa discrepância.

Por outro lado, uma métrica de desempenho entra em cena após o treinamento, avaliando o quão bem o modelo pode generalizar novos dados e realizar previsões precisas. Essas métricas também são essenciais para comparar diferentes modelos ou configurações e determinar qual obteve o melhor desempenho.

Os itens a seguir destacam as principais distinções entre funções de perda e métricas de desempenho:

#### 1. Otimização vs Avaliação:

- As funções de perda otimizam os parâmetros do modelo durante o treinamento, enquanto as métricas de desempenho avaliam seu desempenho pós-treinamento.

#### 2. Dependência do Modelo:

- Funções de perda dependem da arquitetura e tarefa, enquanto as métricas são menos dependentes, facilitando comparações entre modelos.

#### 3. Minimização vs Maximização:

- O objetivo da *Deep Learning* é minimizar funções de perda. No entanto, métricas visam maximizar (exceto para métricas de erros).

#### 4. Interpretação:

- Funções de perda podem ser desafiadoras de interpretar devido a valores arbitrários. Métricas, por sua vez, são geralmente mais fáceis de entender, sendo aplicáveis em diversas situações.

##### 2.2.2.6.1 Dice Coefficient e *Dice Loss*

As funções de perda comuns incluem *Cross-Entropy Loss*, *Intersection over Union (IoU) Loss*, *Focal Loss*, *Dice Loss*, *Tversky Loss* e *Lovász Loss*. Optou-se por manter a nomenclatura em inglês devido a sua ampla utilização, inclusive na literatura nacional.

Entre as funções de perda, o *Dice Loss* é amplamente empregado em imagens médicas. Proposto por (MILLETARI; NAVAB; AHMADI, 2016), também conhecido como *Dice similarity coefficient*, é utilizado para avaliar a similaridade entre a máscara de segmentação predita e a máscara real. Esta função de perda é definida como:

$$DiceLoss = 1 - \frac{2 \cdot intersection(pred, gt)}{|pred| + |gt|} \quad (2.14)$$

onde:

- $pred$  é a máscara de segmentação predita,
- $gt(groundtruth)$  é a máscara real,
- $intersection(pred, gt)$  é a intersecção entre  $pred$  e  $gt$ ,
- $|pred|$  e  $|gt|$  são o número total de pixels nas máscaras predita e real, respectivamente.

#### 2.2.2.6.2 IoU

As métricas de desempenho comuns incluem *Intersection over Union (IoU) Loss*, *Mean Intersection over Union (mIoU)*, *Pixel Accuracy*, *Average Precision (AP)*, *BF Score* e *Panoptic Quality*.

Entre essas métricas, o *IoU* é frequentemente utilizado em diversos estudos de segmentação semântica. Também conhecido como *Jaccard Loss* ou *Jaccard Index (JI)*, o *IoU* tem como objetivo prever uma máscara de segmentação por pixel para uma dada imagem. Sua definição é dada pela razão entre a intersecção das máscaras predita e real e a união dessas máscaras, conforme a equação:

$$IoU = \frac{1}{n} \sum_{i=1}^n \frac{pred_i \cap gt_i}{pred_i \cup gt_i} \quad (2.15)$$

onde:

- $pred_i$  é a máscara predita por pixel  $i$ ,
- $gt_i$  é a máscara real por pixel  $i$ ,
- $pred_i \cap gt_i$  é a intersecção entre as máscaras predita e real, e
- $pred_i \cup gt_i$  é a união entre as máscaras predita e real.

#### 2.2.2.7 Algoritmos de Otimização

Afirmado por (ANALYTICS VIDHYA, 2023), em *Deep Learning*, otimizadores são algoritmos que ajustam os parâmetros dos modelos durante o treinamento para minimizar a função de perda. Em outras palavras, esses algoritmos capacitam a rede neural a aprender por meio da atualização iterativa dos pesos e vieses.

Esse comportamento tem impacto direto na precisão e na velocidade de treinamento dos modelos de *Deep Learning*. Durante o treinamento do modelo, a cada iteração completa sobre o conjunto de treinamento, os atributos da rede neural, como os pesos e a taxa de aprendizado, são ajustados.

(SUN, 2019) propõe uma decomposição do problema de otimização em três etapas: (1) Iniciar o algoritmo e conduzi-lo à convergência para uma solução razoável; (2) Garantir que o algoritmo alcance a convergência o mais rápido possível; (3) Assegurar que o algoritmo atinja a convergência para uma solução com um valor objetivo baixo.

Assim como em outras áreas do conhecimento, o estudo possui suas próprias terminologias, sendo as mais difundidas: época (*Epoch*), amostra (*Sample*), lote (*Batch*), taxa de aprendizado (*Learning Rate*), função de perda (*Loss Function*), pesos e vieses (*weight/bias*).

- **Época:** Refere-se ao número de vezes que o algoritmo executa sobre todo o conjunto de treinamento.
- **Amostra:** É um conjunto de tamanho determinado do conjunto de dados.
- **Lote:** Denota o número de amostras que é utilizado para atualizar os parâmetros do modelo.
- **Taxa de Aprendizado:** É o parâmetro que fornece ao modelo uma escala de quanto os pesos devem ser atualizados.
- **Função de perda:** Representa a diferença entre o valor predito e o valor real.
- **Pesos e Vieses:** São os parâmetros de aprendizagem do modelo que controlam o sinal entre dois neurônios.

Existem sete componentes-chaves para algoritmos de otimização no contexto de *Machine Learning*:

1. **Função objetiva:** Também conhecida como função de perda, mede a diferença entre a saída predita do modelo e o valor alvo atual, visando minimizar essa função. Pode também maximizar certos tipos de tarefas de aprendizagem por reforço.
2. **Parâmetros:** São variáveis ou pesos ajustados pelo algoritmo de otimização para minimizar ou maximizar a função objetiva, definindo a configuração do modelo de *Machine Learning*.
3. **Processo de otimização:** Envolvendo atualizações iterativas nos parâmetros do modelo com base nos gradientes da função objetiva. Esse processo continua até alcançar um critério de convergência.

4. **Gradientes:** Indicam a direção e magnitude da função mais íngreme da função objetiva. Algoritmos de otimização usam esses gradientes para determinar e ajustar os parâmetros do modelo.
5. **Taxa de aprendizado:** Hiperparâmetro que controla o tamanho dos passos durante cada iteração no processo de otimização, influenciando a velocidade de convergência e a estabilidade do algoritmo.
6. **Algoritmos de otimização populares:** Incluindo nomes comuns na literatura como descida do gradiente, descida do gradiente estocástica (SGD) e variantes mais avançadas como Adam, RMSProp e Adagrad. Cada um desses algoritmos possui características próprias, normalmente ajustados para diferentes problemas específicos.
7. **Crítérios de convergência:** Determinam quando o processo de otimização será interrompido.

#### 2.2.2.7.1 Otimizador Adam

Os desenvolvedores do otimizador Adam, (KINGMA; BA, 2017), definem-no como um algoritmo de otimização de primeira ordem, baseado no gradiente de funções objetivas estocásticas e fundamentado em estimativas adaptativas de momentos de baixa ordem. O nome Adam é uma abreviação de Estimativa de Momento Adaptativo (*Adaptive Moment Estimation*).

O algoritmo incorpora as vantagens de dois métodos distintos: o *Adaptive Gradient* (*AdaGrad*), conhecido por sua eficácia na adaptação a gradientes esparsos, e o *Root Mean Square Propagation* (*RMSProp*), reconhecido pelo tratamento eficiente de ambientes não estacionários. Além disso, destaca-se por sua implementação direta e pelo baixo consumo de memória.

A regra de atualização para o otimizador Adam envolve a manutenção de duas médias móveis para cada parâmetro: o primeiro momentum (média) e o segundo momentum (variância não centrada). Essas médias móveis são então utilizadas de maneira adaptativa para ajustar as taxas de aprendizagem para cada parâmetro durante o treinamento.

Do ponto de vista matemáticos, a regra de atualização para o otimizador Adam pode ser resumida da seguinte forma: Dado um vetor de parâmetros  $\theta$ , um gradiente  $g_t$  em um dado tempo  $t$ , com hiperparâmetros  $\beta_1$  e  $\beta_2$  e uma pequena constante para estabilidade numérica  $\epsilon$ :

1. Inicializa o tempo  $t = 0$ , o primeiro momentum inicial  $m_0 = 0$  e o segundo momentum inicial  $v_0 = 0$ .



2. Atualiza o primeiro momentum estimado:

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \quad (2.16)$$

3. Atualiza o segunda momentum estimado:

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot (g_t)^2 \quad (2.17)$$

4. Corrige os vieses nos momentum primeiro e segundo:

$$m_{bt} = \frac{m_t}{1 - \beta_1^t} \quad (2.18)$$

$$v_{bt} = \frac{v_t}{1 - \beta_2^t} \quad (2.19)$$

5. Atualiza os parâmetros:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{v_{bt}} + \epsilon} \quad (2.20)$$

sendo:

- $\alpha$  é a taxa de aprendizado
- $\beta_1$  e  $\beta_2$  são as taxas de decaimento exponencial para o momentum estimado (tipicamente próximos de 1).
- $t$  são os passos;
- $\epsilon$  é uma pequena constante para prevenir a divisão por zero.

O otimizador Adam destaca-se por sua eficiência, rápida convergência e excelente desempenho de generalização, sendo amplamente empregado em diversas aplicações de *Deep Learning*.

#### 2.2.2.8 Técnica para aumentar a quantidade de dados

Considerando as pesquisas de (MUMUNI; MUMUNI, 2022), aumento de dados (data augmentation) é uma técnica desenvolvida com o objetivo de ampliar o volume, a qualidade e a diversidade dos conjuntos de dados de treinamento.

A expansão da quantidade de dados implica na melhoria do desempenho dos modelos de *Machine Learning*, mesmo quando empregamos algoritmos menos sofisticados. Esses novos dados são gerados a partir de informações já existentes ou criados seguindo regras que façam sentido dentro do contexto da aplicação. Diante disso, este método tem a capacidade de reduzir as exigências sobre algoritmos mais complexos, permitindo que arquiteturas de modelos simples alcancem alto desempenho na generalização.

O problema geral do aumento de dados pode ser formulado conforme segue: dado um conjunto de dados  $D$  consistindo por  $S_i$  amostras de treinamento com rótulos correspondentes  $L_i$ , a tarefa do aumento de dados consiste em compor operações de transformação  $T_j$  que podem ser aplicadas nas amostras originais  $S_i$  para criar dados de treinamento adicionais  $S'_i$  sem alterar os rótulos correspondentes. Essas operações de distorção são conhecidas como operações de transformação que preservam os rótulos (label-preserving transformation operations) no processamento de imagens e na literatura de visão computacional. Portanto, a transformação  $T_j(S_i)$  resulta em uma amostra modificada que ainda pode ser semanticamente descrita pelo rótulo original  $L_i$ .

Diversas operações de transformação podem ser utilizadas para diversificar os dados. No entanto, devido a limitações de recursos computacionais, custo e possíveis efeitos indesejáveis de dados espúrios, geralmente é recomendável selecionar apenas um subconjunto das técnicas mais úteis para a aplicação em questão.

## 3 Metodologia da Pesquisa

A seção “Metodologia” descreve em detalhes a abordagem utilizada para a implementação e avaliação do modelo U-Net na detecção e segmentação do disco óptico. A transparência nos métodos empregados é fundamental para assegurar a confiabilidade e a reprodutibilidade dos resultados obtidos.

### 3.1 Descrição do Modelo U-Net

Na seção 2.2.2.5, apresentamos a U-Net como uma arquitetura de rede neural convolucional composta por dois componentes principais e simétricos: o *Encoder* e o *Decoder*. A operação de dupla convolução, seguida da ativação *ReLU*, incorpora a técnica *BatchNormalization* para melhorar a estabilidade e a velocidade de convergência durante o treinamento, expressa aqui como D-Conv por simplicidade.

A função responsável pelo *Encoder* utiliza a operação de D-Conv seguida de *max-pooling*, resultando em duas saídas: a saída da D-Conv, armazenada para uso posterior no *Decoder*, e a saída *max-pooling*, usada na próxima operação de *Encoder*. Após quatro operações de *Encoder*, a quantidade de mapas de recursos aumenta em 64, 128, 256, e 512, respectivamente, culminando na operação de Bottleneck, que aumenta para 1024 a quantidade de mapas de recursos.

O *Decoder* utiliza convolução transposta, concatenando o resultado com a saída da D-Conv armazenada, seguido de uma nova operação de D-Conv. Quatro operações de *Decoder* diminuem a quantidade de mapas de recursos em 512, 256, 128, 64, respectivamente. A operação final gera a máscara de predição através de uma convolução com ativação sigmóide.

### 3.2 Pré-processamento de Dados

O conjunto de dados ARIA consiste em 143 imagens de fundo da retina colorida, cada uma com uma máscara binária correspondente de vasos sanguíneos e 120 imagens das máscaras binárias do DO com a fóvea. As máscaras do DO com a Fóvea foram separadas usando a biblioteca OpenCV, passando por um preenchimento interno dos contornos para melhorar a capacidade de aprendizado do modelo.

Os dados foram organizados em pastas por grupos, e todas as imagens foram redimensionadas para 512x512 para otimizar o tempo de processamento, mantendo a relevância para o modelo.

### 3.3 Conjunto de Dados

O conjunto de dados, denominado *Automatic Retinal Image Analysis* (ARIA) Dataset, coletado entre 2004 a 2006 no Reino Unido. Para as amostras coletadas, utilizou-se imagens da retina entre homens e mulheres, sendo esses divididos em três grupos: pessoas consideradas saudáveis (grupo controle); pessoas com degeneração macular relacionada à idade (grupo AMD) e pacientes diabéticos (grupo diabético). Entretanto, não é possível se aprofundar sobre os detalhes dos indivíduos da época por carência de informações.

Esse conjunto de dados, em geral, é formado por imagens que possuem as seguintes características: ocupa 432 KB, com dimensões de 768x576, ou seja, 768 pixels de largura e 576 pixels de altura, com resolução, tanto horizontal quanto vertical, de 100 dpi, sendo um arquivo de extensão tif.

No total, são 143 imagens de fundo da retina coloridas, com igual quantidade de máscaras binárias dos vasos sanguíneos e tendo apenas 120 imagens das máscaras binárias do disco óptico em conjunto com a fóvea. Reforça-se que os dois últimos elementos citados estão dentro de um mesmo arquivo de imagem.

Existem três grupos de imagens sendo 23 imagens do grupo AMD com o arquivo denominado “aria\_a\_markup”, 61 imagens do grupo controle com o arquivo denominado “aria\_c\_markup” e 59 imagens do grupo diabético com o arquivo denominado “aria\_d\_markup”, consulte [ARIA-Website](#).

### 3.4 Treinamento do Modelo

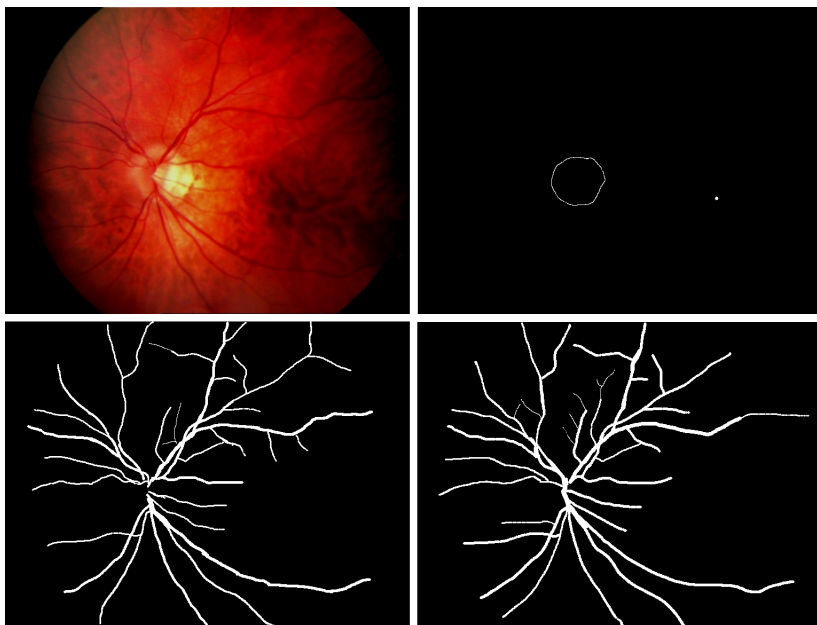
Disponibiliza-se para o modelo 600 imagens para treinamento, sendo 500 delas provenientes de Data Augmentation, utilizando 100 épocas, lotes de tamanho 2 (Devido a quantidade reduzida de dados de treinamento) e taxa de aprendizagem de 0,0001. Os dados foram embaralhados a cada treinamento, evitando que o modelo aprendesse padrões de ordem.

As funções ModelCheckpoint e EarlyStopping do Tensorflow devem ser utilizadas para salvar os melhores parâmetros e interromper o treinamento quando não houver melhorias.

### 3.5 Avaliação do Modelo

O modelo utiliza a função de perda *Dice Loss*, o otimizador *Adam* e métricas de desempenho, incluindo *Dice Coeficiente*, *IoU* e *Pixel Accuracy*. Essas métricas foram detalhadamente exploradas no referencial teórico (consulte a seção 2.2.2.6 para informações

Figura 18 – Da esquerda para a direita, começando de cima e indo para baixo, temos: (1) Imagem colorida da retina; (2) Imagem binária do contorno do DO e o ponto é a localização da fóvea; (3) Imagem binária dos vasos sanguíneos extraídos por profissional da área; (4) Imagem binária dos vasos sanguíneos corrigidos.



Fonte: Imagem (0031)aria\_c\_37 do conjunto de dados ARIA ([FARNELL, 2020](#))

completas). Esses algoritmos estão disponíveis no repositório oficial do Keras, com exceção do otimizador *Adam*, que é proveniente da biblioteca Tensorflow.

Após treinamento, os melhores parâmetros serão salvos para execuções posteriores do modelo em conjunto de dados para predição.

### 3.6 Ferramentas e Ambiente de Desenvolvimento

O ambiente de desenvolvimento adotado é o *Google Colab*, um produto da *Google Research* projetado para facilitar a execução de códigos Python, especialmente em *Machine Learning*, Análise de Dados e Educação. O Colab fornece acesso gratuito a recursos computacionais, incluindo GPU's, sem exigir configuração prévia.

As bibliotecas utilizadas incluem:

- 'os': Funcionalidades dependentes do sistema operacional.
- 'numpy': Fornece um objeto array multidimensional e rotinas para operações com arrays.
- 'cv2': Algoritmos de visão computacional.
- 'tqdm': Exibe o progresso de loops em execução.

- ‘matplotlib’: Cria visualizações estáticas, animadas e iterativas.
- ‘tensorflow’: Constrói e executa gráficos TensorFlow.
- ‘random’: Gera números pseudo-aleatórios.
- ‘glob’: Encontra nomes de caminho correspondentes a um padrão especificado.
- ‘imageio’: Interface fácil para ler e escrever dados de imagem.
- ‘cv2\_show’ do ‘google.colab.patches’: Substitui o método ‘imshow’ do ‘cv2’ desativado no Google Colab.
- ‘albumentations’: Implementa algoritmos de Data Augmentation.
- ‘keras’: Fornece modelos para construção e treinamento de redes neurais.
- ‘sklearn’: Ferramentas para análise de dados e modelagem estatística.

### 3.7 Validação e Teste

Para garantir a generalização dos dados, foram utilizadas métricas de desempenho, incluindo a média do *Pixel Accuracy* e do *IoU*. A comparação das métricas visa verificar a confiabilidade do *IoU* em comparação com o *Pixel Accuracy* nesta aplicação.

### 3.8 Considerações finais sobre a metodologia

Essas seções detalham a metodologia desde a arquitetura do modelo até a seleção do ambiente de desenvolvimento e das bibliotecas utilizadas. Cada etapa foi cuidadosamente considerada para assegurar a clareza, transparência e replicabilidade da pesquisa.

## 4 Desenvolvimento

### 4.1 Implementação do Modelo

A plataforma *TensorFlow* possui um grande conjunto de funções voltadas para *Machine Learning*. Essas funções são o suficiente para implementar o modelo U-Net.

Observe a implementação abaixo:

```

1 from tensorflow.keras.models import Model
2 from tensorflow.keras.layers import Input, Conv2D, Conv2DTranspose, BatchNormalization
3 from tensorflow.keras.layers import Activation, MaxPool2D, Concatenate
4
5 def block_conv(input_tensor, num_filters):
6     x = Conv2D(num_filters, 3, padding="same")(input_tensor)
7     x = BatchNormalization()(x)
8     x = Activation("relu")(x)
9     x = Conv2D(num_filters, 3, padding="same")(x)
10    x = BatchNormalization()(x)
11    x = Activation("relu")(x)
12    return x
13
14 def block_encoder(input_tensor, num_filters):
15    x = block_conv(input_tensor, num_filters)
16    pooled_tensor = MaxPool2D((2, 2))(x)
17    return x, pooled_tensor
18
19 def block_decoder(input_tensor, skip_features, num_filters):
20    x = Conv2DTranspose(num_filters, (2, 2), strides=2, padding="same")(input_tensor)
21    x = Concatenate()([x, skip_features])
22    x = block_conv(x, num_filters)
23    return x
24
25 def model_unet(input_shape):
26    inputs = Input(input_shape)
27    # Encoder
28    s1, p1 = block_encoder(inputs, 64)
29    s2, p2 = block_encoder(p1, 64*2) # 128
30    s3, p3 = block_encoder(p2, 64*4) # 256
31    s4, p4 = block_encoder(p3, 64*8) # 512
32    # Bottleneck
33    bottleneck = block_conv(p4, 64*16) # 1024
34    # Decoder
35    d1 = block_decoder(bottleneck, s4, 64*8) # 512
36    d2 = block_decoder(d1, s3, 64*4) # 256
37    d3 = block_decoder(d2, s2, 64*2) # 128
38    d4 = block_decoder(d3, s1, 64)
39    # Output layer
40    outputs = Conv2D(1, 1, padding="same", activation="sigmoid")(d4)
41    # Create model
42    model = Model(inputs, outputs, name="UNet")
43    return model

```

sendo:

1. Input: Instância um tensor Keras.
2. Conv2D: Camada convolucional 2D.
3. Conv2DTranspose: Camada convolucional transposta.
4. BatchNormalization: Camada que normaliza as entradas.
5. Activation: Aplica uma função de ativação para a saída.
6. MaxPool2D: Operação max pooling global para dados espaciais.
7. Concatenate: Camada que concatena uma lista de entradas.

O código apresenta os dois componentes principais do modelo explicados na seção 2.2.2.5. Dentro da função `model_unet`, encontra-se as quatro operações do *Encoder*, que utilizam da mesma função `block_encoder`, seguindo do Bottleneck e as quatro operações do *Decoder*, que utilizam a mesma função `block_decoder`.

As 23 camadas convolucionais foram mantidas e o funcionamento geral não difere do modelo apresentado, mesmo com a adição do Batch Normalization.

Para as funções de perda e métricas de desempenho:

```

1 from keras import backend as K
2
3 def iou(y_true, y_pred, smooth=1):
4     intersection = K.sum(K.abs(y_true * y_pred), axis=[1, 2, 3])
5     union = K.sum(y_true, axis=[1, 2, 3]) + K.sum(y_pred, axis=[1, 2, 3]) - intersection
6     iou = K.mean((intersection + smooth) / (union + smooth), axis=0)
7     return iou
8
9 def dice_coef(y_true, y_pred, smooth=1):
10    intersection = K.sum(y_true * y_pred, axis=[1, 2, 3])
11    union = K.sum(y_true, axis=[1, 2, 3]) + K.sum(y_pred, axis=[1, 2, 3])
12    dice = K.mean((2. * intersection + smooth) / (union + smooth), axis=0)
13    return dice
14
15 def dice_coef_loss(y_true, y_pred):
16    return 1 - dice_coef(y_true, y_pred)

```

As operações são realizadas conforme explicado na seção 2.2.2.6.1 e seção 2.2.2.6.2.

Reunindo todos os elementos, obtemos o modelo U-Net:

```

1 from tensorflow.keras.optimizers import Adam
2
3 # Defina as dimensões da imagem
4 img_altura = 512
5 img_largura = 512
6
7 # Defina o learning rate
8 lr = 1e-4

```



```

9
10 # Crie o modelo
11 model = model_unet((img_altura, img_largura, 3))
12
13 # Compile o modelo
14 model.compile(
15     loss=dice_coef_loss,
16     optimizer=Adam(learning_rate=lr),
17     # Utilize o argumento 'learning_rate' para especificar o learning rate
18     metrics=[dice_coef, iou, 'accuracy']
19 )
20
21 # Exiba um resumo do modelo
22 model.summary()

```

Adaptou-se a tabela abaixo para organizar o resultado resumido do modelo:

Model: "UNet"			
Layer (type)	Output Shape	Param #	Connected to
input_1	(None, 512, 512, 3)	0	[ ]
conv2d	(None, 512, 512, 64)	1792	[input_1[0][0]]
batch_normalization	(None, 512, 512, 64)	0	[batch_normalization[0][0]]
conv2d_1	(None, 512, 512, 64)	36928	[activation[0][0]]
... (Restante das Camadas)			
conv2d_18	(None, 512, 512, 1)	65	[activation_17[0][0]]
Total params: 31055297 (118.47 MB)			
Trainable params: 31043521 (118.42 MB)			
Non-trainable params: 11776 (46.00 KB)			

Tabela 3 – Resultado do modelo resumido

Neste ponto, o modelo U-Net está apto a ser manipulado para o fim desejado.

## 4.2 Detalhes Técnicos

Os hiperparâmetros como número de camadas, tamanho do filtro, função de ativação, as configurações das camadas, seguem todas as diretrizes do modelo U-Net apresentado na seção 2.2.2.5.

Determinou-se o número de canais (Filtros) para três devido às imagens de entrada possuírem três canais. As alterações de quantidade de mapas de recurso se baseiam no modelo já estudado. O filtro 3x3 utilizado na operação de convolução, a cada novo treinamento, inicia com valores aleatórios e devem melhorar à medida que o treinamento prossegue.

A normalização *Batch Normalization* foi inserida com o objetivo de melhorar o treinamento e a generalização do modelo. A função de perda e otimizador, são baseados

nos estudos das seções 2.2.2.6 e 2.2.2.8. Por fim, a taxa de aprendizagem foi definida conforme uso geral da comunidade de *Machine Learning*.

### 4.3 Tratamento de Dados

O *Google Colab* pode ter acesso ao *Google Drive* executando o seguinte código:

```
1 from google.colab import drive
2
3 # Monta o Google Drive no diretório /content/gdrive
4 drive.mount('/content/gdrive')
```

É necessário certificar-se de autorizar ao *Google Drive* seguindo as instruções fornecidas pelo Colab.

No ambiente de desenvolvimento *Google Colab* as pastas e arquivos criadas ou inseridas pelo usuário são temporárias. Desenvolveu-se, então, um algoritmo que cria diretórios automaticamente no ambiente e outro que carrega as imagens do *Google Drive* para o ambiente.

O *Data Augmentation* foi obtido através da biblioteca *Albumentation*. É necessário instalá-lo no ambiente para utilizar suas funções:

```
1 !pip install -U git+https://github.com/albumentations-team/albumentations
```

Neste ponto o ambiente possui acesso às funções da biblioteca. Neste contexto, foram utilizadas cinco delas:

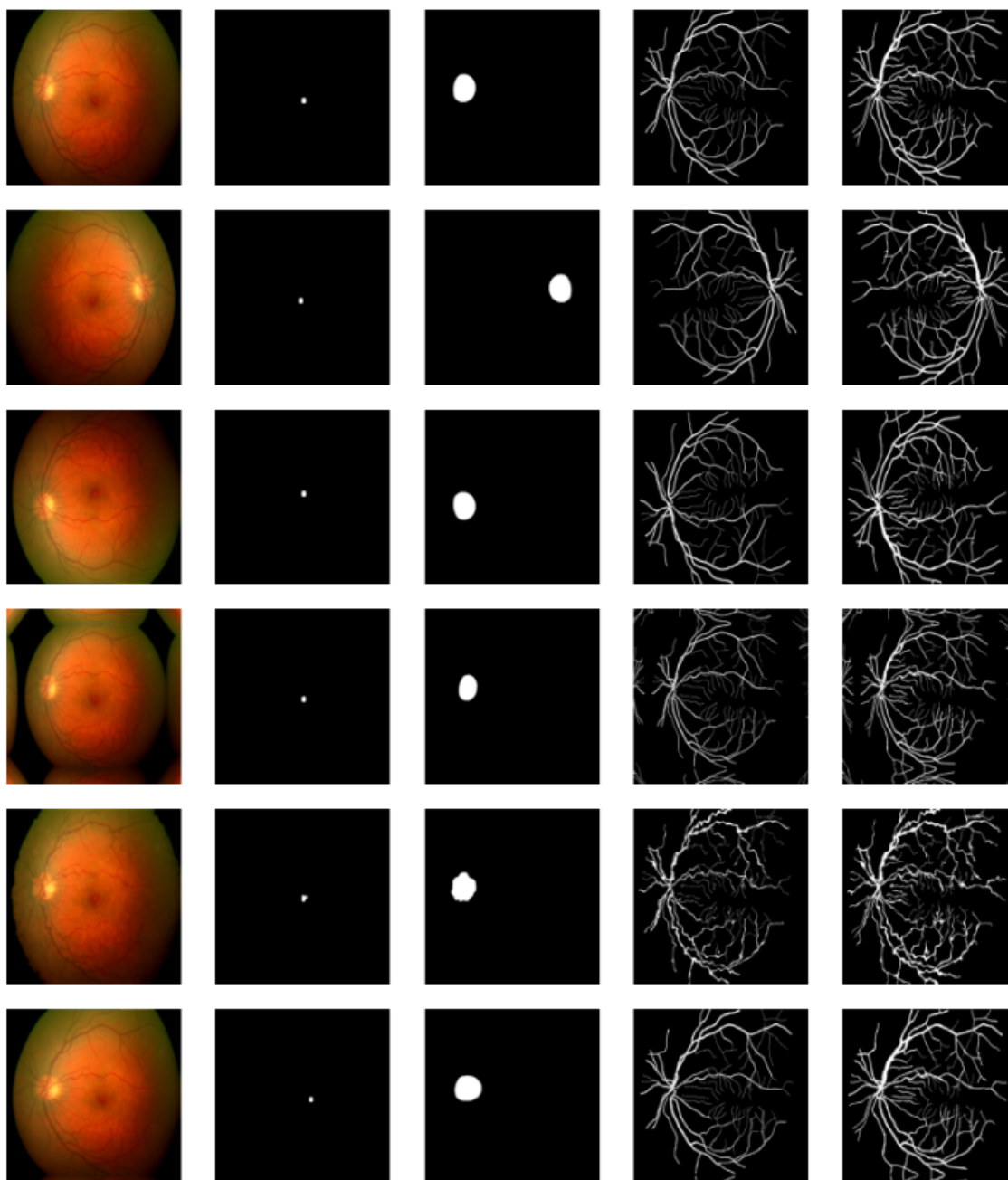
- `HorizontalFlip`: Espelha a imagem horizontalmente;
- `VerticalFlip`: Espelha a imagem verticalmente;
- `OpticalDistortion`: Distorções ópticas;
- `ElasticTransform`: Distorções na estrutura da imagem;
- `GridDistortion`: Distorções na grade da imagem.

As imagens de treinamento, separadas para a aplicação das técnicas, totalizaram 600 imagens de cada tipo após o *Data Augmentation*.

### 4.4 Ferramentas e Bibliotecas

Considerando as bibliotecas que não foram extensamente observadas, segue o resumo de suas contribuições neste trabalho:

Figura 19 – Resultado das operações de data augmentation



Fonte: De autoria própria

- ‘os’: Contribuiu na automação de criação de diretórios e manipulação dos caminhos dos arquivos de interesse.
- ‘numpy’: Facilitou a manipulação das imagens e sua conversão para estruturas compatíveis com o TensorFlow.
- ‘cv2’: Utilizado para leitura das imagens e para salvá-las após o devido tratamento.
- ‘tqdm’: Certas atividades de algoritmos demoram, principalmente quando está lidando com uma grande quantidade de dados. Assim, é possível acompanhar o

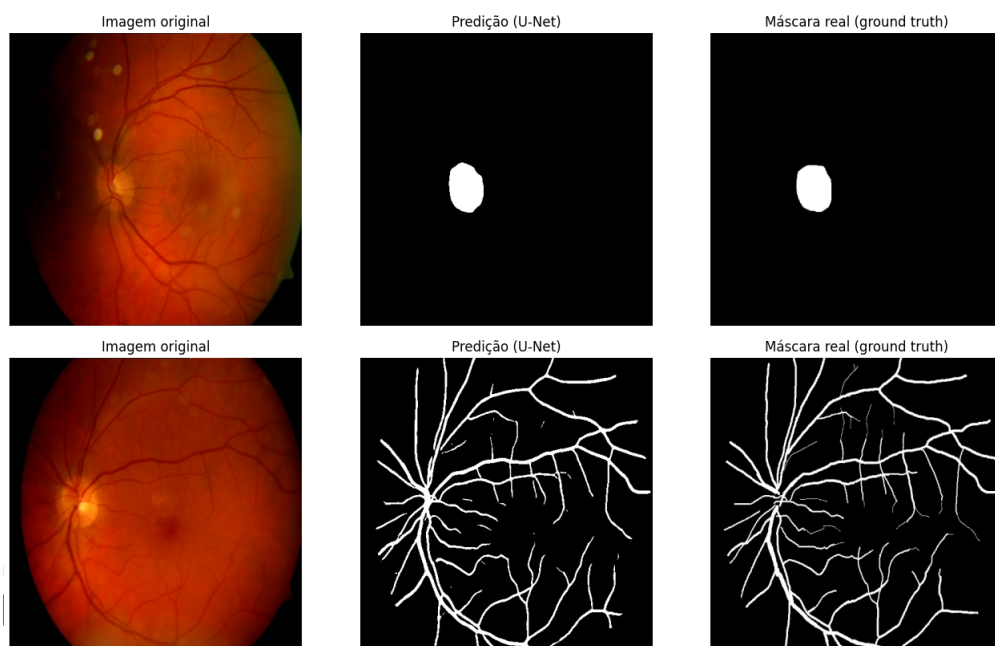
progresso em tempo real do processamento que está sendo executado, conforme exemplo abaixo:

```

1 # Chame a função new_images para diferentes conjuntos de dados
2 new_images(img_train, do_train, fovea_train, vessel_train, 'dataset_final/train/')
3
4 # Saída
5 "100%|██████████|_100/100_[02:52<00:00,___1.73s/it]
```

- ‘matplotlib’: Organizou os resultados, permitindo a comparação visual entre a máscara real e a máscara predita e a visualização do elemento segmentado.

Figura 20 – Utilização do matplotlib no ambiente Colaboratory da Google



Fonte: De autoria própria

- ‘glob’: Considerando que todas as imagens possuem a mesma extensão, permitiu reunir os nomes de todos os arquivos que possuem aquela extensão dentro da pasta, transformado em uma lista para posteriormente ser trabalhada.
- ‘sklearn’: Utilizou-se de sua função ‘accuracy\_score’ para comparação com outra métrica.

## 4.5 Ambiente de Desenvolvimento

A vantagem da utilização do Colab é oferecer, de forma gratuita, os recursos para desenvolver a aplicação de *Machine Learning*. Existe um limite de uso dos recursos, mas para a aplicação em questão foi possível desenvolver a solução completa.

A linguagem de programação utilizada é o Python, atualmente, uma das mais populares, principalmente em atividades de análise de dados. Também o Colab oferece um ambiente que possibilita o desenvolvimento dos códigos em blocos e possibilita a adição de comentários, conforme preferência dos usuários.

Durante o desenvolvimento é importante realizar a configuração do processador utilizado, se é CPU ou GPU. A velocidade de processamento é altamente aprimorada com o uso da GPU, principalmente durante o treinamento do modelo, porém, como mencionado anteriormente, existe um limite no seu uso.

## 4.6 Testes e Validação

Os códigos são avaliados em tempo real, pois a plataforma Colab permite a execução e validação em tempo real. Os modelos fornecidos por meio do *TensorFlow* implementam em sua função de treinamento as avaliações realizadas, necessitando apenas passar o tipo de função de perda e métrica de desempenho utilizadas.

A avaliação final do modelo verifica a média dos resultados obtidos. Neste trabalho utilizou-se duas métricas: *Pixel Accuracy* e *IoU*.

## 4.7 Problemas e Soluções

Para soluções mais robustas, existem versões pagas da plataforma Colab para se obter maiores quantidades de recursos. A quantidade de treinamentos diários realizados limitou-se por volta de 10 execuções.

Deve-se também considerar a quantidade reduzida de dados utilizados. Para aplicações em *Machine Learning* que normalmente aplicam quantidades de dados na casa das dezenas de milhares em diante, a quantidade de recursos fornecidos na versão gratuita não seria o suficiente para permitir o treinamento satisfatório do modelo.

Considerando o exposto, o modelo utilizado poderia alcançar maiores pontuações em suas avaliações com um banco de dados robusto. Com apenas 600 imagens de treinamento de cada tipo após a aplicação do *Data Augmentation*, o modelo não alcançou o nível de generalização desejada, principalmente quando se tratava das poucas imagens que a coloração do disco óptico se mesclava aos demais elementos da imagem.

Entretanto, mesmo considerando a baixa quantidade de imagens, é notável a eficiência do modelo em se ajustar a diferentes problemas, aprendendo com os seus padrões e não necessitando de um operador para realizar modificações frequentes ou complexas.

## 4.8 Resultados Intermediários

A tabela a seguir, apresenta a evolução das métricas ao longo de cinco épocas durante o treinamento do modelo:

Epoch	Loss	Dice Coef	Iou	Acc	Val Loss	Val Dice Coef	Val IoU	Val Acc	Time/ Epoch
1	0.8060	0.1940	0.1125	0.8918	0.9474	0.0526	0.0278	0.9838	02:52
2	0.6165	0.3835	0.2460	0.9821	0.5757	0.4243	0.2726	0.9815	02:24
3	0.4399	0.5601	0.4069	0.9882	0.4761	0.5239	0.3763	0.9914	02:39
4	0.3337	0.6663	0.5249	0.9894	0.2639	0.7361	0.5951	0.9931	02:39
5	0.2468	0.7532	0.6296	0.9914	0.1676	0.8324	0.7183	0.9948	02:38

Tabela 4 – Resultado do Treinamento

As métricas incluem o *Dice Loss (Loss)*, *Dice Coefficient*, *Jaccard Index (IoU)*, *Pixel Accuracy (Acc)* e as métricas correspondentes no conjunto de validação. O tempo por época também é registrado. Esses dados podem ser úteis para avaliar o desempenho do modelo ao longo do tempo e identificar possíveis melhorias.

O treinamento do modelo alcançou melhorias até 35 épocas e encerrou sua execução após 10 épocas sem melhorias. Para salvar o treinamento que obteve melhorias significativas utilizou-se a função `ModelCheckpoint`. Para encerrar a execução, utilizou-se a função `EarlyStopping` configurada para 10 épocas consecutivas. Os parâmetros refinados são armazenados no Google Drive para uso posterior.

A máscara de predição pode ser observada na Fig. 21. A Fig. 22 apresenta o resultado da segmentação do DO. Por fim, a avaliação do modelo utilizou as média da *IoU* e a média da *Accuracy*, conforme Fig. 23.

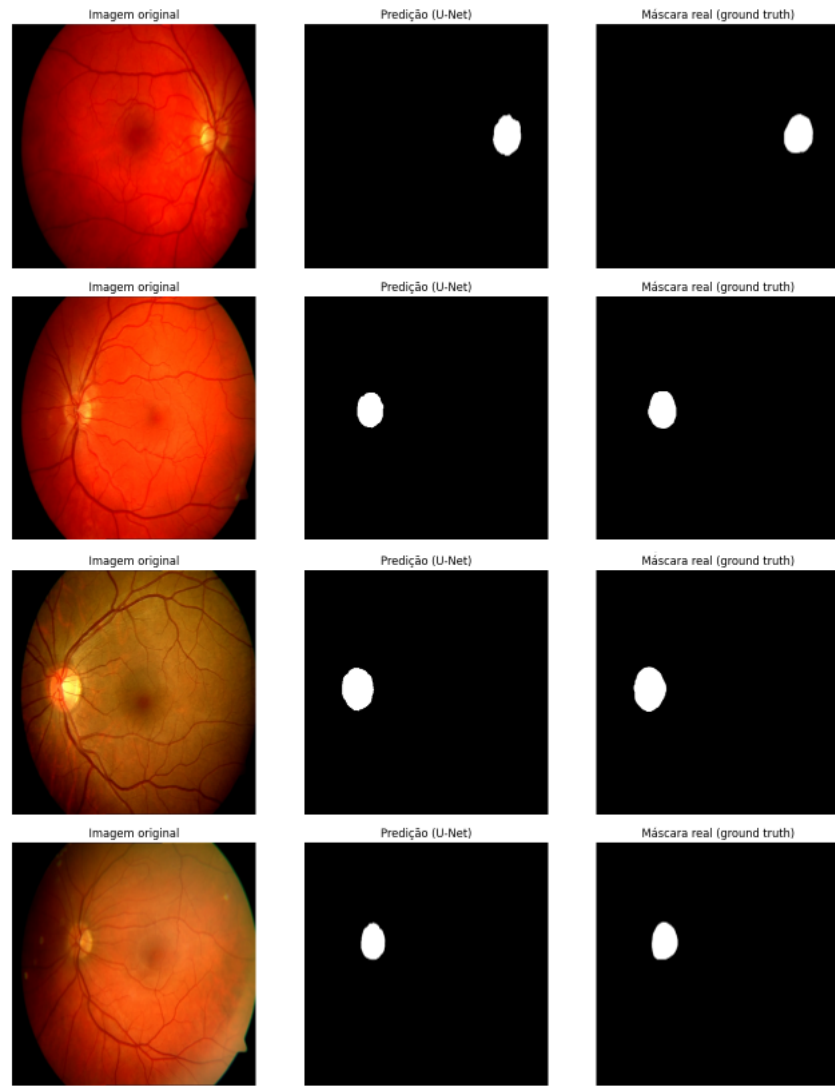
Entre as 20 imagens de teste analisadas, a predição atingiu uma média de 90,56% quando baseado na *IoU* e uma média de 99,70% quando baseado na *Accuracy*.

## 4.9 Iterações e Refinamentos

Quando o tema é *Machine Learning*, existe uma comunidade inteira se mobilizando para desenvolver soluções cada vez mais criativas e eficientes para alcançar resultados cada vez mais surpreendentes.

A leitura e a busca por novas soluções foram o suficiente para acrescentar as modificações necessárias e visualizar o funcionamento do modelo. A cada dia, novos métodos são criados e no momento pesquisadores estão trabalhando para tentar organizar os modelos, as funções de perda e os otimizadores em categorias compreensíveis unificando o conhecimento.

Figura 21 – Da esquerda para a direita, a imagem da retina, a máscara predita pelo modelo e a máscara real



Fonte: De autoria própria

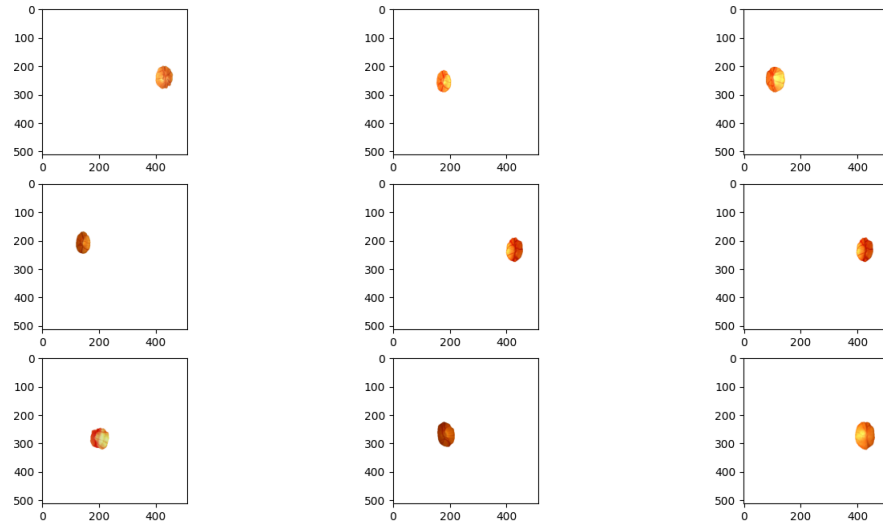
## 4.10 Conexão com a Metodologia

A implementação do modelo se baseou na estrutura do U-Net sendo suficiente apenas seguir os passos de sua criação. Para isso o *TensorFlow* foi a ponte para montar o modelo a partir de suas funcionalidades e permitir acoplar, de forma intuitiva, as funções de perda e métricas escolhidas.

O ambiente de desenvolvimento foi fundamental para elaborar a aplicação, dando as ferramentas necessárias para a sua execução. Além disso, o tratamento das imagens possibilitou melhorias da generalização do modelo, mesmo com uma quantidade pequena de dados.

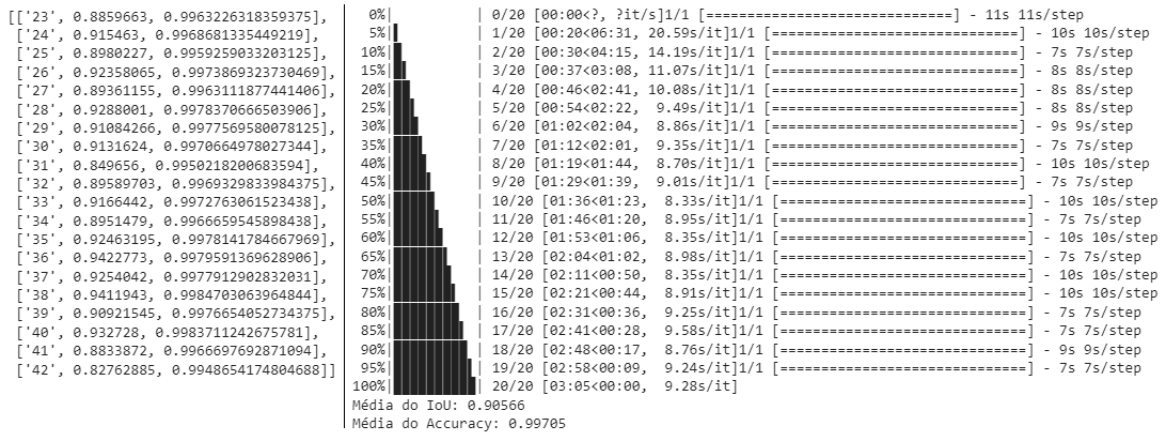
Não houve a necessidade de realizar alterações no modelo, sendo a metodologia

Figura 22 – Resultado da segmentação utilizando as máscaras previstas



Fonte: De autoria própria

Figura 23 – Avaliação do modelo



Fonte: De autoria própria

confiável o suficiente para alcançar a segmentação semântica do DO e observando que extrapolou sua função segmentando, também, os vasos sanguíneos.



## 5 Discussão

### 5.1 Avaliação do Desempenho do Modelo

O poder de generalização do modelo é notável, mesmo considerando a limitada quantidade de imagens na base de dados. A função de perda apresentou resultados abaixo de 0,01, enquanto a métrica de desempenho atingiu uma taxa de acerto de 90,56%. Esses indicadores destacam a eficácia do modelo.

Ao segmentar os elementos, o modelo demonstrou resultados promissores conforme comparações abrangentes dos métodos de segmentação de disco óptico apresentados em (CARDOSO, 2019), (SALLOW; SAHIKHA, 2019), (SAYADIA et al., 2018), além da vantagem de exigir a mínima interferência humana. Isso libera tempo para que o pesquisador se dedique a aspectos mais críticos que demandam maior atenção.

### 5.2 Identificação de Problemas

Apesar do desempenho sólido, identificou-se que a precisão do modelo poderia ser aprimorada com um conjunto de dados mais robusto. A escassez de exemplos limitou o número de épocas de aperfeiçoamento durante o treinamento. A expansão do conjunto de dados, no entanto, requer mais tempo e possivelmente recursos adicionais, sendo a versão gratuita do Colab um limitador nesse contexto.

### 5.3 Interpretação de Resultados

Os resultados obtidos são notáveis, mas para aplicações práticas, especialmente onde a precisão é crucial para diagnósticos precisos, o modelo requer refinamento para alcançar taxas de precisão mais elevadas. Como ferramenta de apoio a profissionais experientes, o modelo demonstra utilidade em seus propósitos.

### 5.4 Análise Visual

Para aqueles não familiarizados com oftalmologia ou sem treinamento específico em reconhecimento de elementos em imagens de retina, os resultados podem parecer praticamente idênticos. No entanto, até profissionais da área são surpreendidos pela velocidade, precisão e capacidade de processamento simultâneo do modelo.

## 5.5 Possíveis Melhorias

Conforme discutido na seção 2.2.2.8, o aumento da quantidade de dados permite o uso de algoritmos menos complexos, mantendo a precisão. Profissionais de saúde e da computação estão colaborando para obter conjuntos de dados mais abrangentes e modelos mais robustos, juntamente com funções de perda e otimizadores mais poderosos.

## 5.6 Impacto nos Objetivos do Projeto

Considerando o prazo da pesquisa e a inexperiência inicial, os resultados são satisfatórios para a finalidade de segmentação automática do DO. O projeto não apenas alcançou seus objetivos, mas também forneceu insights valiosos para direções futuras na pesquisa em segmentação de imagens médicas.

## 6 Conclusão

O propósito fundamental deste estudo foi a segmentação do disco óptico, revelando sua importância nos diagnósticos oftalmológicos. A pesquisa proporcionou uma compreensão abrangente da estrutura ocular, concentrando-se na camada mais interna para análise detalhada de cada componente, aprimorando a compreensão desses elementos.

No âmbito da computação, as imagens da retina tornam-se objetos de análise em disciplinas como Processamento Digital de Imagens (PDI). Nesse contexto, técnicas de Segmentação são empregadas para realçar artefatos de interesse, facilitando análises subsequentes.

A complexidade da segmentação pode apresentar desafios na adaptação de algoritmos a problemas específicos, destacando a necessidade de autonomia dos agentes. Nesse sentido, a pesquisa avançou na área de Inteligência Artificial, culminando na escolha da U-Net, uma rede neural convolucional, reconhecida por seu desempenho na segmentação semântica em imagens médicas. A compreensão da estrutura do modelo, composta por *Encoder* e *Decoder*, foi essencial.

A avaliação dos modelos treinados baseou-se na função de perda *Dice Loss* e na métrica *IoU*, otimizando com o popular algoritmo *Adam*. O resultado de 90,56% de precisão na segmentação, embora não suficiente para aplicações práticas, supera muitos algoritmos clássicos, mostrando-se satisfatório em relação aos objetivos propostos.

Este estudo introduz conceitos importantes em segmentação, destacando o modelo U-Net e resumindo soluções existentes. Dada a inexperiência inicial, limitamo-nos a pesquisar e aplicar técnicas para detecção e segmentação de imagens retinianas, tornando-se uma referência valiosa para iniciantes. A falta de um conjunto de dados robusto, entretanto, limitou as generalizações do modelo.

Diante do crescimento exponencial das inovações em *Machine Learning*, os resultados complexos e amplamente aplicáveis carecem de classificação e análise adequada. Nesse contexto, trabalhos que organizam ideias principais se tornam guias essenciais, evitando abordagens confusas e facilitando a compreensão do tema.

# Referências

- AGGARWAL, C. C. *Neural Networks and Deep Learning: A Textbook*. [S.l.]: Springer; 1st ed., 2018. Citado 2 vezes nas páginas 28 e 32.
- ANALYTICS VIDHYA. *imagem*. 2023. Disponível em: <<https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/>>. Acesso em: 20 nov. 2023. Citado na página 45.
- AZEVEDO-MARQUES, P. M. de; MENCATTINI, A.; SALMERI, M.; RANJAYAN, R. M. *Medical Image Analysis and Informatics: Computer-Aided Diagnosis and Therapy*. [S.l.]: CRC Press, 2018. Citado na página 16.
- BOWLING, B. *Kanski Oftalmologia Clínica: Uma abordagem sistêmica 8ª edição*. [S.l.]: GEN Guanabara Koogan, 2016. Citado na página 18.
- CARDOSO, C. de Fátima dos S. *Segmentação automática do disco óptico e de vasos sanguíneos em imagens de fundo de olho [recurso eletrônico]*. Tese (Doutorado) — Universidade Federal de Uberlândia, 2019. Citado 2 vezes nas páginas 13 e 64.
- CLARO, M. L.; VERAS, R.; SANTOS, L.; FRAZAO, M.; FILHO, A. C.; LEITE, D. Métodos computacionais para segmentação do disco óptico em imagens de retina: uma revisão. *Revista Brasileira de Computação Aplicada*, v. 10, n. 2, p. 29–43, 2018. Citado 2 vezes nas páginas 12 e 13.
- COPPIN, B. *Inteligência Artificial*. [S.l.]: LTC; 1ª edição, 2010. Citado na página 34.
- DICIONARIO PRIBERAM. *imagem*. 2023. Disponível em: <<https://dicionario.priberam.org/imagem>>. Acesso em: 15 nov. 2023. Citado na página 22.
- ERWIN, E.; CANIA, B.; LARASATI, K. Technique automatic for detection and segmentation of optic disc area in retinal image. *Journal of Physics: Conference Series*, v. 1500, p. 012089, 04 2020. Citado na página 13.
- FARNELL, D. J. *Automated Retinal Image Analysis (ARIA) Data Set*. 2020. Disponível em: <[http://www.damianjffarnell.com/?page\\_id=276](http://www.damianjffarnell.com/?page_id=276)>. Acesso em: 10 jan. 2022. Citado 2 vezes nas páginas 14 e 52.
- GONZALEZ, R. C.; WOORS, R. E. *Processamento Digital de Imagens 3ª Edição*. [S.l.]: Pearson Prentice Hall, 2010. Citado 5 vezes nas páginas 12, 22, 23, 24 e 25.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: The Mit Press, 2016. Citado na página 28.
- HOS. *Exame ocular que pode detectar doenças no corpo*. 2016. Disponível em: <<https://www.hosergipe.com.br/blog/exame-ocular-que-pode-detectar-doencas-no-corpo/>>. Acesso em: 24 out. 2021. Citado na página 12.
- KINGMA, D. P.; BA, J. *Adam: A Method for Stochastic Optimization*. 2017. Citado na página 47.

LAVINNE, J. B.; M.D. *The Eye Care Sourcebook*. [S.l.]: McGraw Hill; 1st edition, 2001. Citado na página 17.

LENAIL, A. *NN-SVG*. 2023. Disponível em: <<https://alexlenail.me/NN-SVG/index.html>>. Acesso em: 10 dez. 2023. Citado 4 vezes nas páginas 30, 33, 39 e 41.

MCGRAW-HILL COMPANIES. *Chapter 2: Ophthalmologic Examination*. 2023. Disponível em: <[http://www.oculist.net/others/ebook/generalophthal/server-java/arknoid/amed/vaughan/co\\_chapters/ch002/ch002\\_figure13.html](http://www.oculist.net/others/ebook/generalophthal/server-java/arknoid/amed/vaughan/co_chapters/ch002/ch002_figure13.html)>. Acesso em: 10 dez. 2023. Citado na página 19.

MICHAELIS. *imagem*. 2023. Disponível em: <<https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/imagem/>>. Acesso em: 15 nov. 2023. Citado na página 22.

MILLETARI, F.; NAVAB, N.; AHMADI, S.-A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: *2016 Fourth International Conference on 3D Vision (3DV)*. [S.l.: s.n.], 2016. p. 565–571. Citado na página 44.

MINISTERIO DA SAUDE. Vigitel brasil 2019: vigilância de fatores de risco e proteção para doenças crônicas por inquérito telefônico: estimativas sobre frequência e distribuição sociodemográfica de fatores de risco e proteção para doenças crônicas nas capitais dos 26 estados brasileiros e no distrito federal em 2019[recurso eletrônico]. In: \_\_\_\_\_. [S.l.]: Secretaria de Vigilância de Saúde, 2020. Citado na página 12.

MITCHELL, T. M. *Machine Learning*. [S.l.]: McGraw-Hill Science/Engineering/Math, 1997. Citado na página 27.

MUMUNI, A.; MUMUNI, F. Data augmentation: A comprehensive survey of modern approaches. *Array*, v. 16, p. 100258, 2022. ISSN 2590-0056. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2590005622000911>>. Citado na página 48.

NIELSEN, M. A. *Neural Networks and Deep Learning*. [S.l.]: Determination Press, 2015. Citado 4 vezes nas páginas 28, 31, 32 e 36.

PIXEON. *O que é CAD*. 2014. Disponível em: <<https://www.pixeon.com/blog/o-que-e-cad/>>. Acesso em: 24 out. 2021. Citado na página 12.

PRAKASH, N.; GR, H.; M, M. An efficient approach on optic disc detection and retinal blood vessel segmentation for diabetic retinopathy based on support vector machine. v. 9, p. 1031–1038, 01 2016. Citado na página 13.

RIORDAN-EVA, P.; WHITCHER, J. P. *Oftalmologia Geral de Vaughan & Asbury 17ª edição*. [S.l.]: AMGH Editora Ltda, 2011. Citado 4 vezes nas páginas 12, 16, 18 e 20.

RONNEBERGER, O.; FISCHER, P.; BROX, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. Citado 2 vezes nas páginas 41 e 42.

RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. [S.l.]: Prentice Hall; 3rd ed., 2009. Citado 4 vezes nas páginas 26, 27, 28 e 36.

- SALLOW, A. B.; SAHIKHA, H. K. Optical disc and blood vessel segmentation in retinal fundus images. *Academic Journal of Nawroz University*, v. 8, n. 3, 2019. Citado 2 vezes nas páginas 13 e 64.
- SANNIHIT; PURI, S.; SINGH, S. Image segmentation techniques: A survey. *International Journal of Engineering and Applied Physics (IJEAP)*, 05 2021. ISSN 2737-8071. Citado 2 vezes nas páginas 24 e 25.
- SAYADIA, S. B.; ELLOUMI, Y.; AKIL, M.; HEDIBEDOUI, M. Computational Efficiency of Optic Disk Detection on Fundus Image: A survey. In: *SPIE Real-Time Image and Video Processing*. Orlando, United States: [s.n.], 2018. Disponível em: <<https://hal-upec-upem.archives-ouvertes.fr/hal-01796769>>. Citado 2 vezes nas páginas 13 e 64.
- SIEGFRIED, D. R. *Anatomia e fisiologia Para Leigos*. [S.l.]: Alta Books; 1ª edição, 2010. Citado 2 vezes nas páginas 17 e 18.
- SOCIEDADE BRASILEIRA DE CARDIOLOGIA. Arquivos brasileiros de cardiologia: 7ª diretriz brasileira de hipertensão arterial. In: \_\_\_\_\_. [S.l.]: Revista da Sociedade Brasileira de Cardiologia, 2016. v. 107, n. 3. Citado na página 12.
- SUN, R. Optimization for deep learning: theory and algorithms. *CoRR*, abs/1912.08957, 2019. Disponível em: <<http://arxiv.org/abs/1912.08957>>. Citado na página 46.
- TERVEN, J.; CORDOVA-ESPARZA, D.-M.; RAMIREZ-PEDRAZA, A.; CHAVEZ-URBIOLA, E. Loss functions and metrics in deep learning. a review. 07 2023. Citado na página 43.
- VIVA OFTALMO. *10 doenças nos olhos mais comuns que você deve conhecer*. 2020. Disponível em: <<https://vivaofthalmologia.com.br/doencas-dos-olhos/>>. Acesso em: 24 out. 2023. Citado na página 19.
- ZHANG, Z.; SRIVASTAVA, R.; LIU, H.; CHEN, X.; DUAN, L.; WONG, D. W. K.; KWOH, C. K.; LIU, T. Y. W. . J. A survey on computer aided diagnosis for ocular diseases. *BMC Medical Informatics & Decision Making*, v. 14, n. 80, 2014. Citado na página 13.