Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia Eletrônica

Comparação de Arquiteturas de Deep Learning para Segmentação de Imagens Dermatoscópicas de Melanoma

Autor: Antonio Prado da Silva Júnior, Diogo Gomes de Sousa Bezerra, Yasmine Silveira Andrade

Orientador: Dr. Renan Utida Barbosa Ferreira

Brasília, DF

2020

Antonio Prado da Silva Júnior, Diogo Gomes de Sousa Bezerra, Yasmine Silveira Andrade

Comparação de Arquiteturas de Deep Learning para Segmentação de Imagens Dermatoscópicas de Melanoma

Monografia submetida ao curso de graduação em (Engenharia Eletrônica) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia Eletrônica).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Dr. Renan Utida Barbosa Ferreira

Brasília, DF

2020

Antonio Prado da Silva Júnior, Diogo Gomes de Sousa Bezerra, Yasmine Silveira Andrade

Comparação de Arquiteturas de Deep Learning para Segmentação de Imagens Dermatoscópicas de Melanoma/ Antonio Prado da Silva Júnior, Diogo Gomes de Sousa Bezerra, Yasmine Silveira Andrade. – Brasília, DF, 2020-

82 p. : il. (algumas color.) ; 30 cm.

Orientador: Dr. Renan Utida Barbosa Ferreira

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2020.

1. Deep Learning. 2. Melanoma. I. Dr. Renan Utida Barbosa Ferreira. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Comparação de Arquiteturas de Deep Learning para Segmentação de Imagens Dermatoscópicas de Melanoma

CDU 02:141:005.6

Antonio Prado da Silva Júnior, Diogo Gomes de Sousa Bezerra, Yasmine Silveira Andrade

Comparação de Arquiteturas de Deep Learning para Segmentação de Imagens Dermatoscópicas de Melanoma

Monografia submetida ao curso de graduação em (Engenharia Eletrônica) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia Eletrônica).

Trabalho aprovado. Brasília, DF, 16 de dezembro de 2020:

Dr. Renan Utida Barbosa Ferreira
Orientador

Dr. Bruno Macchiavello
Convidado 1

Dr. Camilo Dorea
Convidado 2

Brasília, DF
2020

Agradecimentos

A Deus por ter nos dado saúde e força para continuar, apesar das adversidades encontradas nesse momento atípico vivido.

Aos amigos, Gabriel Araújo, Nicholas Barros, William Thalisson e demais amigos do SAMU por sempre estarem presentes incentivando e participando de forma direta e indireta em nossas vidas acadêmicas e pessoais no decorrer da nossa formação.

Aos nossos pais e familiares que nos deram todo suporte físico e emocional ao longo de nossa vida e especialmente durante os anos de graduação.

A todos os professores e em especial, ao nosso orientador Renan Utida, que sempre se mostrou solícito e encorajador durante todo o desenvolvimento do presente trabalho.

Resumo

No Brasil, o câncer de pele representa cerca de 33% dos diagnósticos dentre os tipos de câncer, sendo apenas 3% causados pelo melanoma. Entretanto, esse tipo de câncer possui a maior taxa de mortalidade dentre os cânceres de pele, cerca de 7%. Por tratar-se de uma doença com alta taxa de mortalidade, o diagnóstico precoce do melanoma em seu estágio inicial é essencial para um prognóstico positivo da doença. Devido aos avanços tecnológicos, novos métodos de diagnósticos de doenças de pele estão sendo desenvolvidos para auxiliar profissionais médicos, como por exemplo o diagnóstico auxiliado por computador que utiliza técnicas de aprendizado de máquina e suas ramificações. A segmentação é um dos passos mais importantes do diagnóstico auxiliado por computador, pois acaba afetando a precisão das etapas seguintes. Este trabalho tem como objetivo comparar diferentes técnicas de segmentação de imagens baseadas em aprendizado profundo para segmentação de melanoma em imagens dermatoscópicas. Os *backbones* DenseNet-121, Resnet-50 e VGG-19 foram utilizados na etapa de *encoder* da U-Net para a realização do processo de segmentação. As arquiteturas foram treinadas e testadas utilizando o *dataset* ISIC 2017 com e sem a utilização da técnica de aumento de dados a fim de avaliar o impacto desta técnica nas métricas obtidas. Posteriormente, após a obtenção do modelo treinado, o mesmo foi testado no *dataset* PH^2. Todo o processo de implementação deste trabalho foi feito no ambiente computacional Google Colab em sua versão Pro, utilizando o TensorFlow e o Keras como as principais bibliotecas na implementação das arquiteturas. A arquitetura U-Net + ResNet-50 apresentou índice *Jaccard* de 81.94%, melhor índice nas médias obtidas no *dataset* ISIC com a utilização do aumento de dados, porém, o melhor modelo obtido foi apresentado pela arquitetura DenseNet-121 com 83.64% utilizando o *dataset* $ISIC_A$ com aumento de dados.

Palavras-chaves: Melanoma. Segmentação. Redes Neurais. Aprendizado de Máquina.

Abstract

In Brazil, skin cancer represents about 33% of diagnoses among types of cancer, with only 3% caused by melanoma, however, this type of cancer have a higher mortality rate among skin cancers, about 7%. As it is a disease with a high mortality rate, the early diagnosis of melanoma in its initial stage is essential for a positive prognosis of the disease. Due to technological advances, new methods of diagnosing skin diseases are being developed to assist medical professionals, such as the diagnosis aided by a computer using machine learning techniques and their ramifications. Segmentation is one of the most important steps of computer-aided diagnosis, as it ends up affecting the accuracy of the following steps. This work aims to compare different image segmentation techniques based on deep learning for melanoma segmentation in dermoscopic images. The DenseNet-121, Resnet-50 and VGG-19 backbones were used in the U-Net encoder stage to perform the segmentation process. The architectures were trained and tested using the ISIC 2017 dataset with and without the use of the data augmentation technique in order to assess the impact of this technique on the obtained metrics. Subsequently, after obtaining the trained model, it was tested on the PH^2 dataset. The entire process of implementing this work was done in the Google Colab computing environment in its Pro version, using TensorFlow and Keras as the main libraries in the implementation of the architectures. The U-Net + ResNet-50 architecture presented a Jaccard index of 81.94%, the best index in the averages obtained in the ISIC dataset with the use of data increase, however the best model obtained was presented by the DenseNet-121 architecture with 83.64% using the $ISIC_A$ dataset with data augmentation

Keywords: Melanoma. Segmentation. Neural networks. Machine Learning.

Lista de ilustrações

Figura 1 – Subtipos clínicos de melanoma cutâneo. (A) melanoma expansivo superficial. (B) melanoma nodular (C) LMM (D) Melanoma amelanótico. Fonte: (KLEBANOV et al., 2019) 26
Figura 2 – Representação da vizinhança entre pixels. 28
Figura 3 – Ilustração do processo de segmentação semântica. Fonte: Adaptado de (JORDAN, 2018) 30
Figura 4 – Inteligência Artificial, Aprendizado de Máquina e Aprendizado Profundo. Fonte: Adaptado de (CHOLLET, 2018) 31
Figura 5 – Componentes de um neurônio biológico. Fonte: Adaptado de (MULLER; REINCHARDT; STRICKLAND, 1995) 33
Figura 6 – Neurônio artificial 34
Figura 7 – Rede Neural 34
Figura 8 – Rede Feedforward única. Fonte: Autores 35
Figura 9 – Rede Feedforward de múltiplas camadas. Fonte: Autores 36
Figura 10 – Rede Recorrente. Fonte: Adaptado de (GURNEY, 2004) 36
Figura 11 – Rede Recorrente com neurônios ocultos. Fonte: Adaptado de (GURNEY, 2004) 37
Figura 12 – Função de perda. Fonte: Adaptado de (CHOLLET, 2018) 38
Figura 13 – Função de perda para medir qualidade da saída da rede. Fonte: Adaptado de (CHOLLET, 2018) 38
Figura 14 – Utilização da pontuação de perda para balanceamento dos pesos nas camadas. Fonte: Adaptado de (CHOLLET, 2018) 39
Figura 15 – Rede Neural profunda para reconhecimento de dígitos. Fonte: Adaptado de (CHOLLET, 2018) 40
Figura 16 – Representações profundas aprendidas por um modelo de classificação de dígitos. Fonte: Adaptado de (CHOLLET, 2018) 40
Figura 17 – Camadas típicas de uma CNN. Fonte: Adaptado de (MATHWORKS, 2020) 41
Figura 18 – Processo de correlação de uma matriz 4x4 por um *kernel* 2x2, *stride* de 1 e sem preenchimento. Fonte: (KHAN et al., 2018) 42
Figura 19 – Algumas funções não lineares comuns em arquiteturas de aprendizado profundo. Fonte:(KHAN et al., 2018) 44
Figura 20 – Processo de agrupamento máximo com uma janela 2x2 e um passo de 1. Fonte:(KHAN et al., 2018) 44
Figura 21 – Exemplo de um processo *upsample* utilizando o método de *unpooling*. Fonte: Adaptado de (YIN; YAN; SHIN, 2019) 45

Figura 22 – Camada totalmente conectada com função *softmax*. Fonte:(MISSINGLINK.AI, 2020) 47
Figura 23 – Fonte: Adaptado de (SRIVASTAVA et al., 2014) 47
Figura 24 – Exemplo de uma arquitetura LeNet. Fonte: Adaptado de (LECUN et al., 1998) 48
Figura 25 – Arquitetura VGGNet. Fonte: Adaptado de (MUHAMMAD et al., 2018) 49
Figura 26 – Bloco residual de uma ResNet. Fonte: Adaptado de (HE et al., 2016) . 50
Figura 27 – Arquitetura ResNet com 34 camadas residuais. Fonte: Adaptado de (HE et al., 2016) 50
Figura 28 – Bloco denso de 5 camadas e taxa de crescimento $k = 4$, em que $x0, x1, x2$ e $x3$ são camadas densas. Fonte: Adaptado de (HUANG et al., 2017) 51
Figura 29 – Arquitetura DenseNet. Fonte: Adaptado de (HUANG et al., 2017) . . . 51
Figura 30 – Arquitetura U-Net. Fonte: Adaptado de (RONNEBERGER; FISCHER; BROX, 2015) 53
Figura 31 – Ilustração do processo de concatenar mapas de características, assim como ocorre em uma U-Net. 53
Figura 32 – Matriz de confusão. 54
Figura 33 – Exemplos de algumas transformações aplicadas as imagens dermatoscópicas, e suas respectivas máscaras, para o aumento de dados. (a) imagem original, (b),(c), (d) transformações. 61
Figura 34 – Funções de perda CE e FL, para valores variados de γ e $\alpha = 1$. Fonte: Adaptado de (Lin et al., 2017) 66
Figura 35 – Gráficos das perdas durante o treinamento para os *datasets* de treino e validação. Eixo vertical corresponde ao valor da perda e o eixo horizontal a época. 72
Figura 36 – Máscaras de segmentação preditas pelos modelos treinados com o *dataset* $ISIC_A$, utilizando imagens do *dataset* ISIC como teste, para as respectivas arquiteturas, manchas e GT (do inglês, *Ground Truth*) ou Verdade Fundamental. 73
Figura 37 – Máscaras de segmentação preditas pelos modelos treinados com o *dataset* $ISIC_A$, utilizando imagens do *dataset* PH^2 como teste, para as respectivas arquiteturas, manchas e GT 73

Lista de tabelas

Tabela 1 – Resultados das arquiteturas construídas por Jahanifar et al. 56
Tabela 2 – Resultados da arquitetura construída por Sheng Chen et al. 57
Tabela 3 – Transformações utilizadas no segundo método de aumento de dados. . 60
Tabela 4 – Comparação entre CPUs das máquinas virtuais 62
Tabela 5 – Comparação de bibliotecas usadas para Aprendizado Profundo 63
Tabela 6 – Hiperparâmetros utilizados. 64
Tabela 7 – Métricas U-Net+DenseNet-121 69
Tabela 8 – Métricas U-Net+ResNet-50 69
Tabela 9 – Métricas U-Net+VGG-19 69
Tabela 10 – Comparação entre a média das métricas do *dataset* ISIC 70
Tabela 11 – Métricas obtidas para o *dataset* PH^2 70
Tabela 12 – Comparação de resultados de arquiteturas encontradas na literatura . . 74

Lista de abreviaturas e siglas

AC	Acurácia
AI	Inteligência Artificial
ANN	Redes Neurais Artificiais
CAD	Diagnóstico Auxiliado Por Computador
CE	Entropia cruzada
CNN	Rede Neural Convolucional
DenseNet	Rede convolucional densamente conectada
DL	Aprendizado Profundo
ELM	Microscopia de epiluminescência
EP	Especificidade
FCN	Rede Totalmente Convolucional
FL	Perda focal
FN	Falso Negativo
FP	Falso Positivo
GT	Verdade Fundamental
ILSVRC	ImageNet Desafio de Reconhecimento Visual em Grande Escala
INCA	Instituto Nacional do Câncer
IoU	Intersecção na União
ISDIS	International Society for Digital Imaging of the Skin
ISIC	Nome do desafio para análise de lesão de pele para detecção de melanoma
L	Função de perda
LMM	Melanoma Lentigo maligno
MLP	Redes Feedforward de Múltiplas Camadas

$N_4(p)$	Relação entre pixels chamada de vizinhança-4
$N_8(p)$	Relação entre pixels chamada de vizinhança-8
P	Precisão
ReLU	Unidade Linear Retificadora
ResNet	Rede Residual
RGB	Padrão de imagem que possuem 3 camadas de cores: vermelho, verde e azul
RNN	Redes Neurais Recorrente
ROI	Região de Interesse
SE	Sensibilidade
SSM	Melanoma de Disseminação Superficial
TN	Verdadeiro Negativo
TP	Verdadeiro Positivo
UV	Raios Ultravioleta

Lista de símbolos

α	Pesos de ponderação
γ	Parâmetro de foco

Sumário

1 INTRODUÇÃO 21
1.1 Contextualização 21
1.2 Problema de pesquisa 21
1.3 Justificativa 23
1.4 Objetivos 23
1.4.1 Objetivos Gerais 23
1.4.2 Objetivos Específicos 23
1.5 Estrutura do Texto 23

2 REFERENCIAL TEÓRICO 25
2.1 Câncer de Pele 25
2.2 Melanoma 25
2.2.1 Melanócitos 25
2.2.2 Classificação Clínica e Histológica do Melanoma 26
2.3 Imagens Digitais 28
2.3.1 Filtragem 29
2.3.2 Segmentação Semântica 29
2.4 Inteligencia Artificial 30
2.5 Aprendizado de Máquina 31
2.6 Redes Neurais 32
2.6.1 Estruturas de Redes Neurais 34
2.6.2 Treinamento 37
2.7 Aprendizado Profundo 39
2.8 Redes Neurais Convolucionais - CNN 40
2.8.1 Camada Convolucional 41
2.8.2 Não-linearidade 43
2.8.3 Camada de agrupamento 43
2.8.4 Camada de *upsample* 45
2.8.5 Camada totalmente conectada 46
2.8.6 *Dropout* 46
2.9 Arquiteturas de Redes Neurais Convolucionais 47
2.9.1 LeNet 48
2.9.2 VGG 49
2.9.3 ResNet 49
2.9.4 DenseNet 50

2.9.5 U-Net . . . 52
2.10 Métricas de Desempenho . . . 52
2.11 Trabalhos Relacionados . . . 55
2.11.1 Jahanifar et al. (2018) . . . 55
2.11.2 Sheng Chen et al (2018) . . . 56

3 METODOLOGIA . . . 59
3.1 Base de Dados . . . 59
3.2 Aumento de dados . . . 60
3.3 Arquiteturas Utilizadas . . . 61
3.4 Recursos Computacionais . . . 62
3.4.1 *Hardware* . . . 62
3.4.2 *Software* . . . 63
3.5 Implementação . . . 64
3.5.1 U-Net . . . 64
3.5.2 Hiperparâmetros . . . 64
3.5.3 Função de Perda . . . 64
3.6 Métricas de Desempenho . . . 66
3.7 Procedimento de teste . . . 67

4 RESULTADOS E DISCUSSÕES . . . 69

5 CONCLUSÕES E PROJETOS FUTUROS . . . 75
5.1 Trabalhos futuros . . . 75

REFERÊNCIAS . . . 77

1 Introdução

1.1 Contextualização

De acordo com os dados emitidos pela OMS (Organização Mundial de Saúde) no ano de 2020, a aparição de câncer de pele não melanoma anualmente está entre 2 e 3 milhões de casos. Já para os casos de melanoma, estima-se cerca de 132 mil casos em todo o mundo (OMS, 2020). No Brasil, o câncer de pele representa cerca de 33% dos diagnósticos dentre os tipos de câncer, sendo apenas 3% causados pelo melanoma (INCA, 2020a). Entretanto, esse tipo de câncer possui a maior taxa de mortalidade dentre os cânceres de pele, cerca de 75% (CONTE, 2019). Estima-se mais de 180 mil novos casos no território nacional em 2020, sendo que 8.450 correspondem ao melanoma. Além disso, a taxa de mortalidade do câncer de pele apresenta crescimento ano após ano, com 20% de mortalidade no ano 2000 e 31% no ano de 2018 (INCA, 2020a).

O câncer de pele melanoma é mais comum em pessoas brancas do que em pretas. Estatisticamente, a chance de contrair a doença no decorrer da vida é de 2,6% em pessoas brancas e 0,1% em pessoas pretas (ACS, 2020a). A idade média das pessoas diagnosticadas com a doença é de 65 anos, entretanto o melanoma é um dos cânceres mais comuns entre jovens e adultos, especialmente em mulheres (ACS, 2020a). Apesar da incidência da doença ser maior em homens, antes dos 50 anos as taxas são maiores em mulheres (ACS, 2020b).

Por tratar-se de uma doença com alta taxa de mortalidade, o diagnóstico precoce do melanoma em seu estágio inicial é essencial para um prognóstico positivo da doença. Isso pode ser quantificado através da taxa de sobrevida relativa de 5 anos, a qual estima a possibilidade de uma pessoa que contraiu a doença estar viva após 5 anos, dependendo do estado do diagnóstico da doença (localizado, regional ou distante) (ONCOGUIA, 2020). Para o estágio inicial do melanoma (localizado) a taxa de sobrevida relativa é de 99%, enquanto que para outros estágios, onde o câncer já se espalhou para nódulos linfáticos (estágio regional) e outros órgãos (estágio distante), as taxas de sobrevida são 65% e 25%, respectivamente (ACS, 2020b).

1.2 Problema de pesquisa

Diversas técnicas de imagens não invasivas são utilizadas para auxiliar o profissional responsável na realização do diagnóstico. Imagens por ressonância magnética, ultrasom, dermatoscopia e imagens espectroscópicas são alguns exemplos de tais técnicas

(SMITH; MACNEIL, 2011). Imagens macroscópicas, mais conhecidas como imagens clínicas, e imagens adquiridas por microscopia de epiluminescência (ELM, do inglês epiluminescence microscopy), também chamadas de dermatoscopia ou imagens de dermatoscopia (ZHOU et al., 2010) são normalmente usadas na análise computacional de lesões cutâneas. Para análises feitas utilizando imagens clínicas a precisão do diagnóstico fica em torno de 60%, já com a utilização das imagens de dematoscopia, essa taxa varia de 75% à 85% (H.KITTLER et al., 2017).

Mesmo a dermatoscopia apresentando uma melhora de 10% a 30% na sensibilidade diagnóstica (MAYER, 1997), conforme S. Menzies et al. (MENZIES et al., 2005) existe uma grande diferença no diagnóstico de lesões cutâneas realizado por diferentes profissionais em relação a suas habilidades e especialidades. O estudo mostrou que profissionais especializados na área dermatoscopica foram capazes de alcançar 90% de sensibilidade e 59% de especificidade no diagnóstico, enquanto profissionais menos especializados, como clínicos gerais, obtiveram 62% de sensibilidade e 63% de especificidade.

Devido aos avanços tecnológicos, novos métodos de diagnósticos de doenças de pele estão sendo desenvolvidos para auxiliar profissionais médicos, como por exemplo o diagnóstico auxiliado por computador, ou CAD (do inglês *computer-aided diagnosis*) (SILVEIRA et al., 2009). O CAD é um sistema que auxilia os médicos na interpretação de imagens médicas. É composto de 3 etapas: 1) segmentação da imagem; 2) extração de características e 3) classificação da lesão. A segmentação é um dos passos mais importantes do CAD, pois acaba afetando a precisão das etapas seguintes. Entretanto, a segmentação apresenta dificuldades referentes a sua execução devido aos elementos presentes nas imagens dermatoscopicas (SILVEIRA et al., 2009). A variedade de tamanhos, cores, texturas e presença de pelos sobre a lesão são apenas alguns dos obstáculos encontrados nas imagens que acaba afetando o resultado da segmentação.

A segmentação, tem como objetivo atribuir um rótulo de categoria a cada *pixel* de uma imagem, o que é uma tarefa fundamental na pesquisa de visão computacional (HAO; ZHOU; GUO, 2020). Em geral, a segmentação baseia-se na análise de propriedades de uma região de interesse, ou ROI (do inglês *Region of Interest*) a partir de suas similaridades e descontinuidades, existindo diversas técnicas para a análise desses tipos de características. Segmentações baseadas em limites, bordas, regiões, contornos ativos e AI (do inglês *Artificial Intelligence*) são algumas dessas técnicas (OLIVEIRA et al., 2016).

Outra abordagem utilizada para a realização da segmentação, é a utilização de algoritmos baseados em aprendizado de máquina, ou ML (do inglês *machine learning*). Entre os métodos existentes de aprendizado de máquina, o aprendizado profundo, ou DL (do inglês *deep learning*) vem apresentando resultados significativos para reconhecimento de padrões em imagens médicas (JAFARI et al., 2016). De acordo com Han et al. (HAN et al., 2018) o DL é um ramo das arquiteturas de aprendizado de maquina, que tenta

modelar abstrações de alto nível a partir do processamento em camadas utilizando redes neurais (ou do inglês, *neural networks*).

1.3 Justificativa

Um diagnóstico rápido e preciso é necessário para que o tratamento precoce do câncer melanoma aumente as chances de sobrevivência do paciente. Com o intuito de agilizar o processo de detecção, ferramentas computacionais relacionadas a AI estão se tornando cada vez mais recorrentes na área de análise de imagens clínicas. Mesmo com diversas técnicas e ferramentas, ainda é possível obter melhores resultados nos índices que classificam a precisão da segmentação realizada nas manchas de pele, melhorando os resultados do CAD. Nesse contexto, viu-se a necessidade de comparar diferentes tipos e arquiteturas de aprendizado profundo para a utilização destas ferramentas no estágio inicial do processo de classificação de imagens clínicas, que é a segmentação da mancha de pele.

1.4 Objetivos

1.4.1 Objetivos Gerais

Comparação de diferentes técnicas de segmentação de imagens baseadas em aprendizado profundo para segmentação de melanoma em imagens dermatoscópicas.

1.4.2 Objetivos Específicos

- Fazer o levantamento de técnicas de aprendizado profundo para segmentação de imagens médicas.
- Selecionar códigos publicamente disponíveis que sejam aplicáveis a imagens de melanoma.
- Selecionar banco (ou bancos) de imagens públicos para testes.
- Fazer os ajustes necessários nos códigos para sua execução com os bancos selecionados.
- Comparar objetivamente as técnicas selecionadas.

1.5 Estrutura do Texto

No Capítulo 2 - Referencial Teórico - trazemos um breve estudo a cerca do melanoma tal como suas causas, tratamentos disponíveis e sua classificação no meio clínico.

Neste mesmo capítulo apresentamos uma revisão bibliográfica com a descrição de técnicas de AI e DL, arquiteturas das redes estudadas e as métricas usadas para a avaliação do desempenho das mesmas. No Capítulo 3 - Metodologia - descrevemos a metodologia em que comparamos diferentes arquiteturas e como se comportam aplicadas em diferentes *datasets*. Descrevemos ainda as ferramentas de *hardware* e *software* utilizadas para a execução deste trabalho e o procedimento realizado para a obtenção das métricas de desempenho e máscaras de segmentação. No Capítulo 4 - Resultados - apresentamos as métricas obtidas ao final do treinamento de cada rede e suas máscaras de segmentação. E por fim, no Capítulo 5 - Conclusões - concluímos apresentando uma análise crítica a cerca do desempenho de cada rede, tal como a proposta de projetos futuros.

2 Fundamentação Teórica

Para entendimento do trabalho realizado, primeiro definiremos o que é e quais as características do melanoma, os conceitos de imagem e filtragem digital, aprendizado de máquina e as arquiteturas que serão testadas para a segmentação do melanoma. Este capítulo possui o objetivo de dar base teórica para esta compreensão.

2.1 Câncer de Pele

Câncer é o nome dado a doenças que têm em comum o crescimento desordenado de células, que invadem tecidos e órgãos, havendo mais de 100 tipos (INCA, 2020a). O câncer de pele corresponde a 33% de todos os diagnósticos desta doença no Brasil. E, a cada ano, cerca de 180 mil novos casos são registrados pelo Instituto Nacional do Câncer (INCA). O mais comum deles é o não-melanoma que, como já mencionado no capítulo anterior, possui uma baixa taxa de letalidade.

Esta doença é provocada pelo crescimento anormal e descontrolado de células presentes na pele. Assim, os diferentes tipos de cânceres de pele são classificados de acordo com as camadas da pele que foram afetadas. Os carcinomas basocelulares e os espinocelulares são os tipos mais comuns. Já o mais letal (e ao mesmo tempo o mais raro) é o melanoma, o mais agressivo câncer de pele (SBD, 2020).

2.2 Melanoma

O melanoma é um câncer que se origina nos melanócitos, células produtoras de melanina. Ele é considerado especialmente perigoso porque tem uma capacidade considerável de se espalhar para outros tecidos do corpo (INCA, 2020b).

Uma pesquisa do Instituto Datafolha divulgada em maio de 2018 – encomendada pela farmacêutica Bristol-Myers Squibb – revelou que 78% da população brasileira entrevistada não sabe o que é melanoma. Os dados são preocupantes pois, como mencionado anteriormente, mesmo sendo um tipo menos comum de tumor de pele, ele é o que mais mata (SANTOS, 2019).

2.2.1 Melanócitos

Segundo Ostrowski et. al (OSTROWSKI; FISHER, 2021a), os melanócitos são células que existem principalmente na camada basal da epiderme, camada mais superficial da pele humana, mas também estão localizadas em outros locais, incluindo a úvea, que é

um conjunto presente nos olhos. O papel principal do melanócito é produzir o pigmento melanina dentro dos melanossomas e transferir essas organelas para os queratinócitos (WU; HAMMER, 2014), o principal tipo de célula do cabelo e da pele (OSTROWSKI; FISHER, 2021b). Os melanossomas são localizados nas área perinucleares (cobertura nuclear) e protegem o núcleo das células contra raios ultravioleta (UV) e irradiação e ainda auxilia no controle de calor.

A cor da pele e do cabelo humanos é determinada pela quantidade e tipo de produção de melanina pelos melanócitos cutâneos e foliculares. Não é o número, a distribuição no local do corpo ou a densidade das células melanócitos em si, mas sim a regulação do processo de melanogênese que deve ser examinada para entender as diferenças fenotípicas nas características pigmentares (STURM, 2009), o que caracteriza a pigmentação constitutiva. Há ainda a pigmentação adaptativa, que se trata da pigmentação induzida por radiação UV (bronzeamento), porém este processo também pode ser ativado em casos de distúrbios de hiperpigmentação da pele (OSTROWSKI; FISHER, 2021a).

Os melanócitos produzem duas formas de pigmento melanina, a eumelanina, responsável pela cor preta e marrom, e a feomelanina, responsável pelas cores vermelha e loira, ambas derivadas da tirosina precursora (PROTA, 1980).

2.2.2 Classificação Clínica e Histológica do Melanoma

Existem quatro variantes principais que classificam o melanoma cutâneo primário: melanoma de disseminação superficial, melanoma nodular, melanoma lentigo maligno (ou LMM, do inglês *Lentigo Maligno Melanoma*), e melanoma lentiginoso acral, os quais podem ser observados na Figura 1 (KLEBANOV et al., 2019).

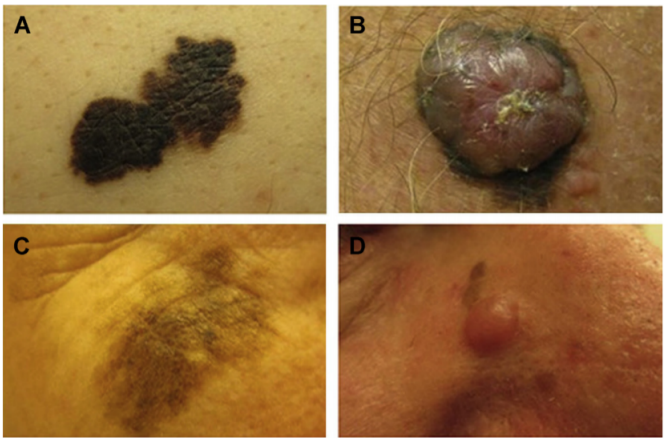


Figura 1 – Subtipos clínicos de melanoma cutâneo. (A) melanoma expansivo superficial. (B) melanoma nodular (C) LMM (D) Melanoma amelanótico. Fonte: (KLEBANOV et al., 2019)

Elas se diferem pelos padrões de crescimento que caracterizam a formação ini-

cial do melanoma nesses grupos, mas o prognóstico é semelhante para todos eles quando normalizado a profundidade de invasão dentro da pele (chamada de profundidade de *Breslow*) no momento do diagnóstico e tratamento (BRESLOW, 1970). Embora a maioria dos melanomas sejam pigmentados, cerca de 5% a 10% dos melanomas cutâneos são amelanóticos. A falta de pigmentação muitas vezes atrasa o diagnóstico, piorando o prognóstico, mas quando normalizado para a profundidade de Breslow, o prognóstico das lesões amelanóticas não difere do melanoma pigmentado (MOREAU; WEISSFELD; FERRIS, 2013).

O melanoma de disseminação superficial (SSM, do inglês *superficial spreading melanoma*) é o tipo mais comum de melanoma (60% - 70% em populações de pele clara). Clinicamente, os SSMs mostram as características marcantes do melanoma ABCD: assimetria, bordas irregulares, variação de cor e diâmetro aumentado (FRIEDMAN; RIGEL; KOPF, 1985). Normalmente, SSMs têm uma fase de crescimento radial prolongada (de meses a anos) caracterizada por uma expansão intraepidérmica sem invasão dérmica que prossegue para um crescimento vertical, fase que está associada à invasão dérmica e pior prognóstico.

Já o melanoma nodular é mais comumente localizado em áreas cronicamente expostas ao sol, como cabeça e pescoço. Na histologia, os melanomas nodulares mostram apenas uma fase de crescimento vertical e acredita-se que ocorram na ausência de uma fase de crescimento radial. Eles crescem rapidamente e geralmente estão presentes em uma profundidade de *Breslow* avançada. Por esse motivo, os melanomas nodulares, que representam apenas 15% a 20% dos melanomas primários, possuem índice de mais de 40% das mortes por melanoma (SHAIKH; XIONG; WEINSTOCK, 2012).

A LMM ocorre principalmente em pacientes idosos por causa da pele cronicamente danificada pelo sol. O LMM é derivado de um precursor do lentigo maligno (LM) *in situ* que se apresenta como uma mácula que aumenta lentamente e muda de marrom para preto com bordas irregulares. Cerca de 5% dos LMs *in situ* progridem para LMM (WEINSTOCK; SOBER, 1987).

Por fim, o melanoma lentiginoso acral, constitui menos de 5% dos cânceres de pele em pessoas brancas, e é o que mais atinge pessoas negras, mas isso ocorre porque a incidência dos outro tipos de melanoma em pessoas de pele escura são mais incomuns (OSTROWSKI; FISHER, 2021a). Não aparenta possuir ligação à incidência do sol e aparece mais frequentemente em palmas, plantas e região subungueal (região abaixo das unhas) (LEÓN et al., 2013). Apresenta clinicamente com mudanças graduais de tamanho, forma e cor, da mesma forma que os SSM e os LMM. Há algumas evidências de que ocorrem em local que sofreram traumas anteriormente e são mais comuns em áreas de estresse mecânico crônico, como por exemplo, em áreas de alta pressão do pé plantar (OSTROWSKI; FISHER, 2021a).

2.3 Imagens Digitais

Uma imagem digital pode ser descrita matematicamente por uma função, *f(x, y)*, onde x e y são coordenadas espaciais finitas do plano cartesiano. Os valores de *f(x, y)* referem-se à intensidade luminosa e é proporcional ao brilho (ou nível de cinza) da imagem naquele ponto (QUEIROZ; GOMES, 2006) (GONZALEZ; WOODS, 2009).

Para o caso de imagens que possuem bandas distintas de frequência, precisa-se de uma função *f(x, y)* para representar cada uma dessas bandas (GONZALEZ; WOODS, 2009). Isso pode ser observado no padrão RGB, comumente utilizado para exibição em telas. As imagens coloridas são formadas através da adição de cores primárias, neste caso, o vermelho (R, do inglês *red*), o verde (G, do inglês *green*) e o azul (B, do inglês, *blue*) (ROCHA, 2010).

Pode-se representar uma imagem por um conjunto matricial $M \times N$, onde M é a altura da imagem e N a sua largura. Os elementos finitos que compõem a imagem digital são comumente chamadas de *pixel* (do inglês, *picture element*) e possuem localização e valores específicos. Algumas relações importantes entre os *pixels* são os de vizinhança. Para facilitar a explicação, considere p e q como dois *pixels* de uma imagem (GONZALEZ; WOODS, 2009).

Um *pixel* p é considerado vizinho de outro *pixel* q quando se eles estiverem na horizontal ou vertical um do outro, podendo ocupar as posições (x+1,y), (x – 1,y), (x, y+1) e (x, y – 1). Por serem 4 possíveis posições, esse conjunto de *pixel* é chamado de *vizinhança-4* e é expresso por $N_4(p)$.

Agora, se q estiverem na diagonal de p - (x+1, y+1), (x+1, y-1), (x–1, y+1) e (x–1, y–1) - eles serão chamados de *ND*, e, juntamente com a *vizinhança-4*, formarão a *vizinhança-8* de $p(N_8(p))$. Na Figura 2.3 é possível observar essa relação, onde os *pixels* representados por cinza claro são a *vizinhança-4* e por cinza escuro são a *vizinhança-8*.

$N_8(p)$	$N_4(p)$	$N_8(p)$
$N_4(p)$	p	$N_4(p)$
$N_8(p)$	$N_4(p)$	$N_8(p)$

Figura 2 – Representação da vizinhança entre pixels.

O campo chamado de *processamento digital de imagens* lida com o processamento de imagens digitais por um computador digital. Seu estudo e desenvolvimento vem sendo crescente em suas duas categorias distintas. A primeira é a manipulação de imagens para interpretação humana, visando o aprimoramento das informações. Já a segunda é focada na análise automática por computador das informações extraídas da imagem. O termo '*processamento de imagens*' é comumente utilizado para representar a primeira categoria, sendo a outra chamada de '*análise de imagens*', '*visão computacional*' ou ainda '*visão por computador*' (FILHO; NETO, 1999).

O processamento de imagens não é algo simples e envolve diversas etapas interconectadas, tendo como início a captura de imagens, que geralmente corresponde à iluminação que é refletida em objetos. A primeira parte que se entende do processamento é o chamado pré-processamento (JAHNE, 1997), podendo envolver etapas como filtragem de ruídos causados pelos sensores de captura da imagem e correção de distorções geométricas (QUEIROZ; GOMES, 2006).

2.3.1 Filtragem

Técnicas de domínio espacial atuam diretamente nos *pixels* de uma imagem, diferente do domínio da frequência, quando as operações são realizadas na transformada de Fourier dessa imagem (GONZALEZ; WOODS, 2009).

A filtragem espacial pode ser utilizada para diversos fins, como é o caso de realces e suavização de imagens. Ela se dá, de modo geral, por uma operação de convolução de uma máscara e da imagem digital considerada. A máscara (também chamada de *kernel*, *mask* ou *template*), por sua vez, pode ser definida por uma matriz, geralmente quadrada, de dimensões inferiores às da imagem a ser filtrada (QUEIROZ; GOMES, 2006).

A convolução espacial é dada pelo processo de rotacionar uma máscara a 180° e movê-la pela imagem (GONZALEZ; WOODS, 2009), calculando a soma dos produtos em cada posição conforme a equação 2.1

$$w(x,y) * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t), \tag{2.1}$$

onde w representa o filtro de tamanho $m \times n$ e f a imagem original.

2.3.2 Segmentação Semântica

A tarefa de segmentação semântica tem como entrada uma imagem digital e tem como objetivo rotular cada pixel dessa imagem para uma determinada classe, preservando a posição espacial de cada pixel classificado na imagem, como pode ser exemplificado na Figura 3. É importante ressaltar que segmentação semântica pode ser dividida em

segmentação binária e multi-classe, referente ao número de classes que se deseja segmentar. Essa tarefa ainda é um desafio mesmo para redes totalmente convolucionais ou FCNs (do inglês *Fully Convolutional Networks*) obtidas por redes neurais convolucionais ou CNNs (do inglês *Convolutional Neural Network*), sendo uma das poucas tarefas em que a análise humana obtém resultados discrepantes em relação ao de tecnologias de visão computacional, como o aprendizado profundo, que será discutido mais a frente (MENZIES et al., 2005).

Figura 3 – Ilustração do processo de segmentação semântica. Fonte: Adaptado de (JORDAN, 2018)

2.4 Inteligencia Artificial

A AI, pode ser definida como o esforço para automatizar tarefas intelectuais realizadas pelos humanos. Ela é um amplo campo que engloba tanto o ML, como o DL, porém não se resume somente a ML e DP, possuindo também abordagens que não dependem de aprendizado (CHOLLET, 2018).

A AI simbólica retrata justamente essa abordagem que independe do aprendizado, onde os programadores realizavam um vasto conjunto de regras explícitas para manipular o conhecimento. Um exemplo dessa AI simbólica são os algoritmos que realizavam partidas de xadrez onde envolviam apenas regras codificadas e elaboradas pelos programadores. Tal abordagem não supria as necessidades relacionadas a problemas que envolviam processos mais complexos como classificar imagens, reconhecimento de fala e tradução de idiomas. A partir de tal necessidade surgiram novas abordagens e uma delas foi o ML (CHOLLET, 2018).

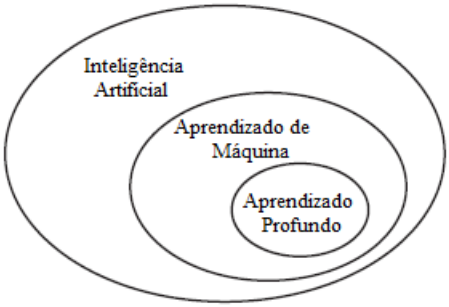


Figura 4 – Inteligência Artificial, Aprendizado de Máquina e Aprendizado Profundo. Fonte: Adaptado de (CHOLLET, 2018)

2.5 Aprendizado de Máquina

O ML é uma técnica usada para analisar e prever padrões a partir de uma amostra de dados para posteriormente ser aplicada em novos dados (NAKAURA et al., 2020). O ML tem como objetivo desenvolver métodos que executem aprendizagem automática utilizando abstrações do mundo real, sem que seja necessário a definição de lógicas por parte dos humanos (KHAN et al., 2018). Segundo Marsland (MARSLAND, 2014) existem diferentes tipos de algoritmos referentes ao aprendizado de máquina, podendo ser classificados em quatro tipos:

- **Aprendizagem supervisionada -** Algoritmos que são treinados a partir de exemplos de respostas corretas (metas). Baseado nesses dados o modelo generaliza as características para responder de maneira correta as entradas possíveis.
- **Aprendizagem não supervisionada -** Não utilizam exemplos de respostas corretas para o treinamento. Ao invés disso, as entradas são analisadas de modo a serem agrupadas por características semelhantes.
- **Aprendizagem por reforço -** Algoritmos que são informados quando a resposta está errada, mas não tem informação de como corrigir tal erro. Isso faz com que o algorítimo teste várias possibilidades até que o erro seja sanado/mitigado.
- **Aprendizagem evolutiva -** Evoluem/mudam de acordo com a ocasião. São implementados seguindo a ideia de aptidão, avaliando quão boa é a solução em determinados momentos.

Conforme Marsland (MARSLAND, 2014) o processo necessário para a implementação do aprendizado de máquina na solução de problemas específicos pode ser descrito pelas etapas a seguir.

- **Coleta e preparação de dados -** Utilização/criação de *datasets* que fornecem dados úteis para o problema em questão, muitas das vezes a maioria dos dados que circundam um problema são úteis, embora possam ser difíceis de serem apanhados, seja devido a quantidade de medições ou a variação de localidade e forma. Para algoritmos de aprendizagem supervisionada, a utilização de gabaritos pode gerar a necessidade de especialistas e uma grande quantidade de tempo. A qualidade e quantidade dos dados utilizados são fatores cruciais para esta etapa.
- **Seleção de recursos -** Ligada à etapa de coleta e preparação de dados, essa etapa tem por finalidade a inspeção dos parâmetros e características mais úteis para o problema em questão, podendo ser necessário conhecimento prévio acerca do problema e dos dados.
- **Escolha de algoritmo -** Seleção do algoritmo baseado em seus princípios básicos, além dos dados em pose.
- **Seleção de parâmetro e modelo -** Grande parte dos algoritmos necessitam da configuração manual de parâmetros, os quais mudam entre diferentes tipos de problemas e soluções, na maioria da vezes, obtidos experimentalmente.
- **Treinamento -** Utilização de recursos computacionais afim de obter um modelo capaz de realizar a predição da saída quando utilizado outros dados de entrada.
- **Avaliação -** Realização de análises e seleção de métricas que qualifiquem e quantifiquem a precisão do modelo treinado.

Inúmeros modelos de algoritmos de aprendizado de máquina foram desenvolvidos no decorrer do tempo, como arvores de decisão, *suport vector machines*, análise de regressão, redes Bayesianas, algorítimos genéticos e redes neurais artificiais (ZHANG; HE; SHAO, 2020). Dentre os modelos de algoritmos de ML as redes neurais são as mais utilizadas atualmente, tendo seu funcionamento inspirado no cérebro humano (KHAN et al., 2018).

2.6 Redes Neurais

Redes neurais artificiais ou ANN (do inglês Artificial Neural Network), comumente chamadas de redes neurais, possuem diversas definições na literatura. Müller et. al (MULLER; REINCHARDT; STRICKLAND, 1995) apresentam uma definição de ANN como algoritmos para tarefas cognitivas que são baseados em conceitos derivados de pesquisas sobre a natureza do cérebro. Segundo Haykin (HAYKIN, 1999), ANN pode ser definida como um processador distribuído maciçamente paralelo composto de unidades de processamento simples que tem uma propensão natural para armazenar conhecimento

experiencial e torná-lo disponível para uso. Gurney (GURNEY, 2004) entretanto, define ANN como um conjunto de elementos de processamento interconectados, onde a capacidade de processamento é armazenada em pesos que são obtidos através de um processo de aprendizado de um conjunto de padrões de treinamento.

Apesar das divergências na definição de ANN, todas as literaturas estudadas apresentam o cérebro como inspiração da criação das ANNs, onde seu principal elemento é o neurônio (Figura 5). Um cérebro humano saudável apresenta cerca de 100 bilhões de células nervosas, conhecidas como neurônios, que se comunicam através de picos curtos de sinais elétricos na membrana celular (GURNEY, 2004).

As junções eletroquímicas, que são responsáveis pelas conexões entre neurônios, são chamadas de sinapse. Tais conexões são realizadas nos ramos dos neurônios, denominados dendritos. Os neurônios realizam diversas conexões com outros neurônios e consequentemente recebem milhares de sinais de entradas que alcançam o corpo da célula. Após o recebimento, os sinais são processados e caso exceda algum limite, o neurônio gera uma impulso elétrico que será transmitido para outros neurônios. Tal transmissão ocorre por meio de uma fibra ramificada conhecida como axônio (MULLER; REINCHARDT; STRICKLAND, 1995).

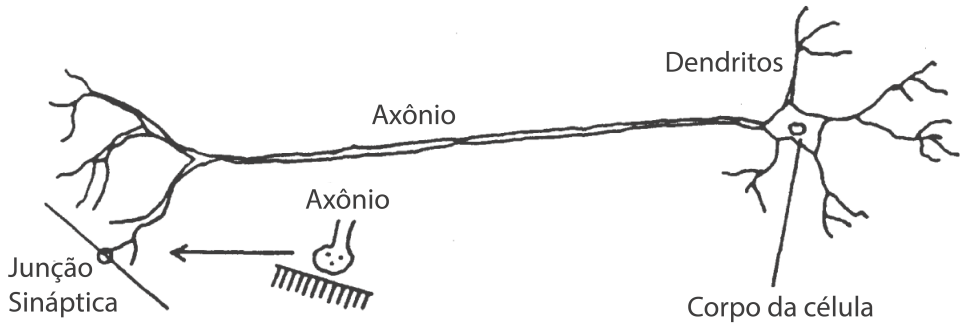


Figura 5 – Componentes de um neurônio biológico. Fonte: Adaptado de (MULLER; REINCHARDT; STRICKLAND, 1995)

O modelo artificial do neurônio biológico é chamado unidade de processamento ou nó. As sinapses nesses modelos são descritas como os pesos aplicados nas entradas, tal que cada entrada seja multiplicada pelo valor do peso antes de ser enviada ao nó. Em seguida, ocorre o processo de soma dos valores recebidos com o intuito fornecer a ativação do nó. Na Figura 6, a ativação é comparada com um valor limiar, caso o valor de ativação ultrapasse esse limiar, o nó produz uma saída de valor alto (1), caso contrário produz uma saída de valor baixo (0) (GURNEY, 2004). Tal configuração pode ser definida como *Perceptron*.

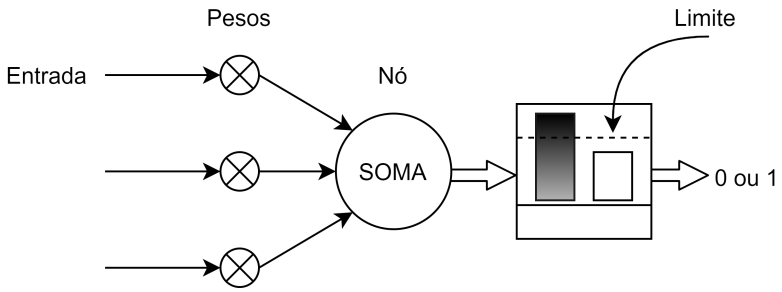


Figura 6 – Neurônio artificial

O termo 'rede' é usado para definir um sistema de neurônios artificiais em que cada um está conectado a todos os outros nós da rede. A Figura 7 representa uma estrutura de rede de neurônios artificiais onde os pesos não são mais presentes, porém são implícitos a cada conexão de entrada em um nó.

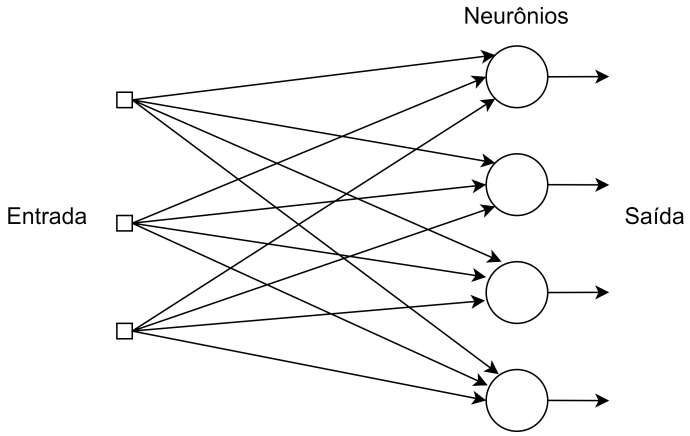


Figura 7 – Rede Neural

2.6.1 Estruturas de Redes Neurais

Em geral é possível identificar três classes de estruturas de redes neurais:

- **Redes *Feedforward* de camada única**

  As formas mais simples de redes em camadas organizam os neurônios em camadas. Nestas redes, a camada de entrada dos neurônios é projetada diretamente para a camada de saída, porém o contrário não acontece. Esse tipo de rede é classificada como estritamente *Feedfoward*, ou seja, o fluxo de informações acontece somente no sentido para frente (HAYKIN, 1999). A Figura 8 representa uma rede *Feedfoward* de camada única, uma vez que a camada de entrada dos nós não é classificada como uma camada propriamente dita, devido ao fato de não ocorrer nenhum tipo de processamento na mesma.

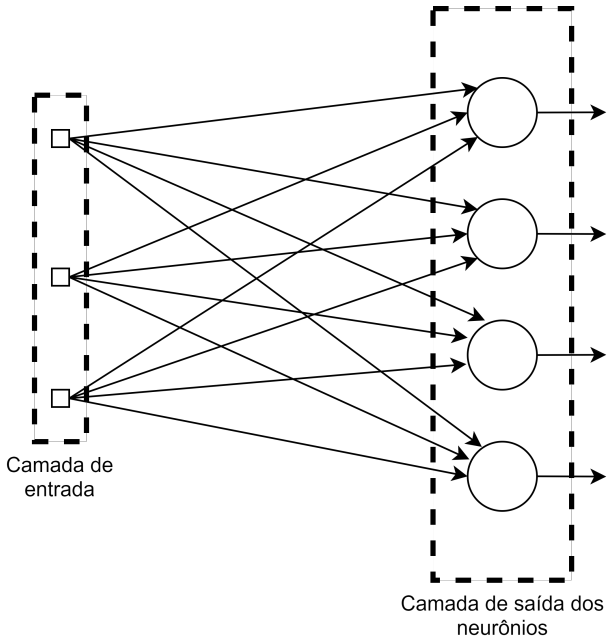


Figura 8 – Rede Feedforward única. Fonte: Autores

- **Redes *Feedforward* de múltiplas camadas**

  Também chamadas de MLPs (do inglês, *Multi Layer Perceptron*), este tipo de estrutura se diferencia da anterior devido a presença de camadas de processamento ocultas. Tais camadas são ditas ocultas por não aparecerem na entrada ou saída da rede. Seu principal objetivo é realizar intervenções úteis entre a entrada e saída da rede. Ao adicionar tais camadas é possível obter um nível maior de abstrações do que as inseridas na entrada da rede (MULLER; REINCHARDT; STRICKLAND, 1995).

  A Figura 9 representa uma rede *feedforward* de múltiplas camadas. As informações inseridas na entrada da camada oculta (segunda camada), provenientes dos sinais de entrada da camada de ativação, produzem uma saída que alimenta a camada seguinte e assim sucessivamente até o final da rede. Geralmente os neurônios em cada camada da rede têm como entradas os sinais de saída apenas da camada anterior. O conjunto de sinais de saída dos neurônios na camada de saída (final) da rede constitui a resposta geral da rede ao padrão de ativação fornecido pelos nós de origem na camada de entrada (primeira)(GURNEY, 2004).

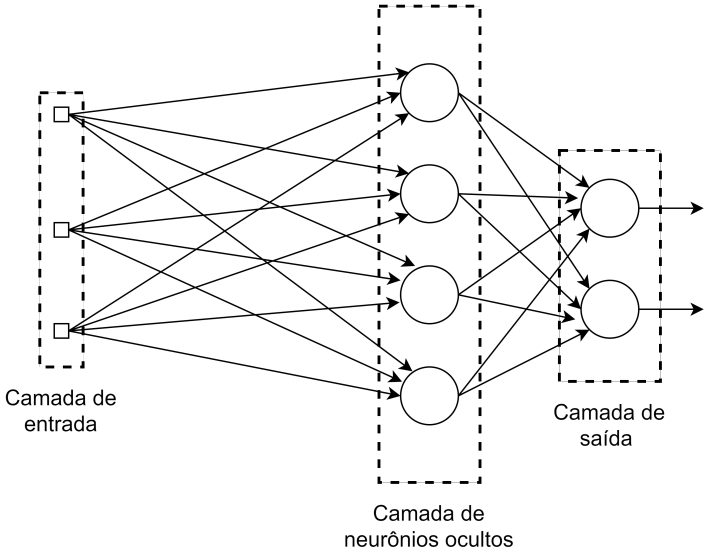


Figura 9 – Rede Feedforward de múltiplas camadas. Fonte: Autores

- **Redes recorrentes**

  As redes recorrentes, ou RNN (do inglês, Recorrent Neural Network) se diferenciam das *feedfowards* por apresentarem pelo menos um *loop* de *feedback*. A Figura 10 apresenta uma rede recorrente com apenas uma camada de neurônios, onde cada neurônio alimenta a entrada de todos os outros da camada a partir de uma realimentação utilizando seu sinal de saída. Tal modelo não apresenta camadas de neurônios ocultos, porém existem redes recorrentes com a presença de neurônios ocultos.

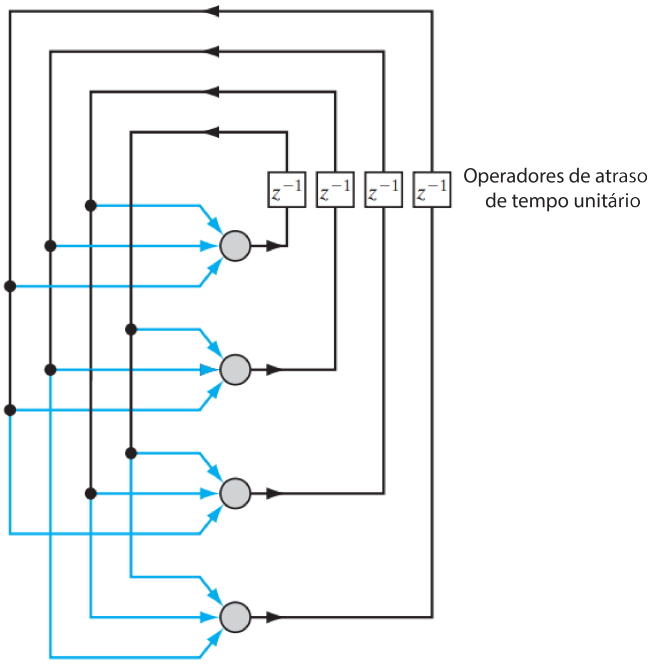


Figura 10 – Rede Recorrente. Fonte: Adaptado de (GURNEY, 2004)

A Figura 11 apresenta uma rede recorrente com neurônios ocultos em que a realimentação provém tanto dos neurônios ocultos, quanto dos neurônios de saída. Tais redes se comportam de forma dinâmica não linear devido a presença presença dos operadores de atraso (Z^{-1}), levando em consideração que a rede não apresente unidades não lineares (GURNEY, 2004).

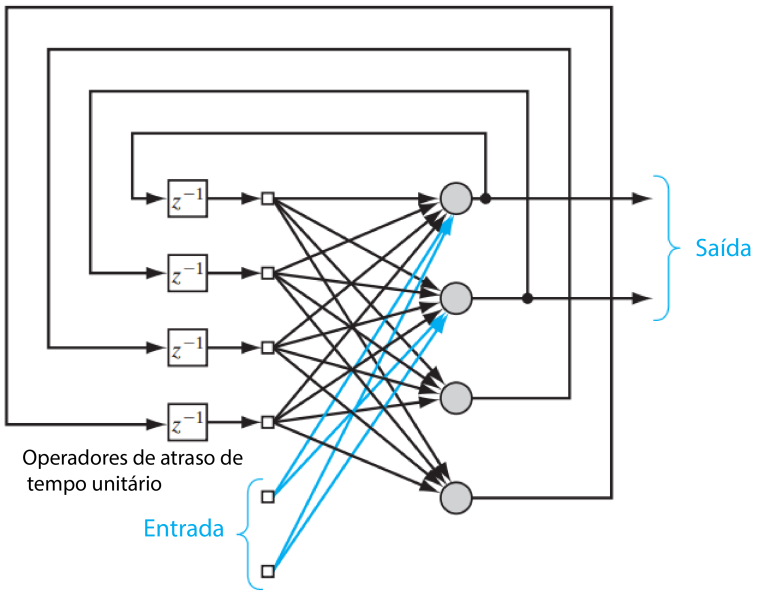


Figura 11 – Rede Recorrente com neurônios ocultos. Fonte: Adaptado de (GURNEY, 2004)

2.6.2 Treinamento

Os resultados da especificação dos dados de entrada obtidos da camada são armazenados nos pesos da camada, também conhecidos como parâmetros da camada. O aprendizado busca justamente encontrar valores para os pesos de todas as camadas que correspondam aos dados de entrada, cada um para seu alvo correspondente. Contudo uma rede neural profunda pode possuir inúmeras camadas e consequentemente vários parâmetros, tornando assim o processo um pouco complexo (CHOLLET, 2018).

O treinamento de ANNs é a etapa responsável pelo aprendizado da rede. Dentro da área de treinamento de ANNs *feedforward*, o algoritmo que possibilita tal procedimento é o *backpropagation*. O *backpropagation* possui duas fases de atuação, o *forward pass* e *backward pass*. O *forward pass* também é conhecido como a etapa de propagação onde as previsões da rede são obtidas através das entradas passadas. Já no *backward pass*, é calculado o valor do gradiente da função de perda para a atualização dos pesos das camadas (CHOLLET, 2018).

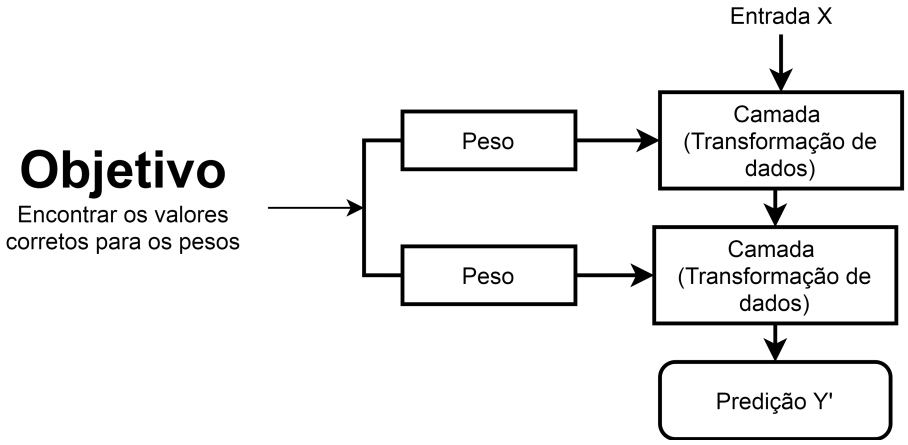


Figura 12 – Função de perda. Fonte: Adaptado de (CHOLLET, 2018)

Para alcançar o resultado desejado no processo de aprendizagem, o controle da saída de uma rede neural é feito observando o quão distante o valor predito está do valor verdadeiro. Tal medida é feita através da função de perda (do inglês *loss function*). A pontuação de perda é obtida através do cálculo da distância entre o valor da predição da rede e o valor verdadeiro esperado (CHOLLET, 2018).

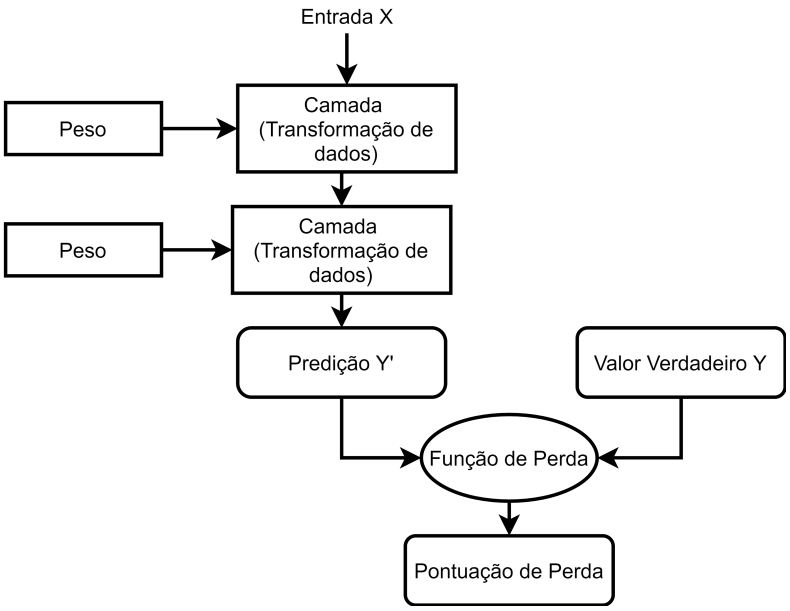


Figura 13 – Função de perda para medir qualidade da saída da rede. Fonte: Adaptado de (CHOLLET, 2018)

A pontuação de perda é utilizada como *feedback* para corrigir os pesos das camadas a fim de obter resultados mais próximos do esperado.

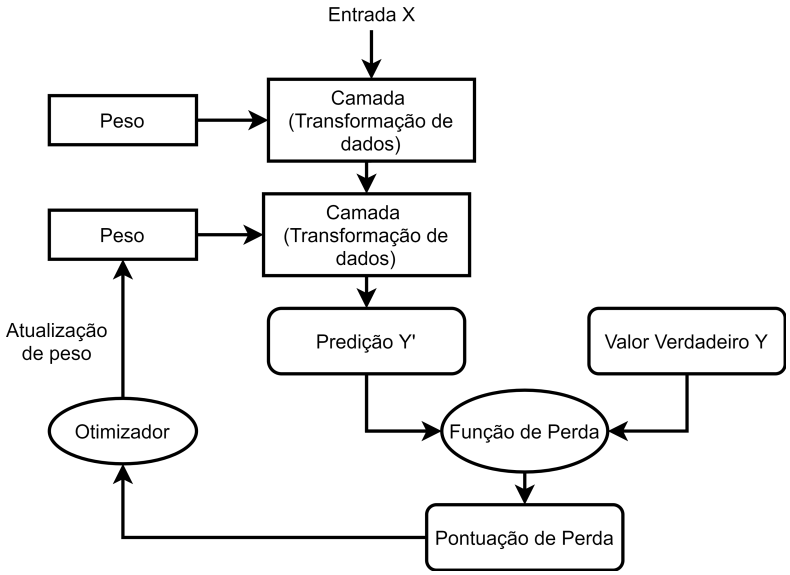


Figura 14 – Utilização da pontuação de perda para balanceamento dos pesos nas camadas. Fonte: Adaptado de (CHOLLET, 2018)

Um dos maiores desafios dentro do treinamento de uma rede neural é justamente evitar que o *overfitting* (ou sobreajuste) ocorra. O *overfitting* ocorre quando o modelo treinado não consegue generalizar bem os dados aprendidos, ou seja, quando é posto diante de dados externos aos usados no treinamento, não consegue ter o desempenho esperado, dando a entender que acabou se ajustando somente aos dados que foram usados no treinamento (KHAN et al., 2018). A regularização por sua vez é uma forma de evitar que o *overfitting* ocorra, onde são aplicadas penalizações, a medida que a complexidade do modelo aumenta, forçando assim que o modelo gerado seja mais simples. As regularizações mais utilizadas são a L1, que penaliza pesos na proporção da soma dos valores absolutos dos pesos, a L2, que penaliza pesos na proporção da soma dos quadrados dos pesos e o *dropout* que por sua vez elimina alguns neurônios/pesos em cada iteração com base em uma probabilidade (VERGARA, 2018).

2.7 Aprendizado Profundo

O DL é uma abordagem dentro do ramo do aprendizado de máquina que utiliza a lógica de camadas das redes neurais, para realizar o processo de aprendizado de forma autônoma (NAKAURA et al., 2020). O DL utiliza representações a partir de camadas sucessivas, enquanto alguns modelos de aprendizado se concentram na utilização de uma ou duas camadas. O DL chega a utilizar dezenas ou até mesmo centenas de camadas sucessivas de representação, todas elas aprendidas automaticamente mediante a exposição aos dados de treinamento (CHOLLET, 2018).

A Figura 15 representa o processo de uma rede neural profunda com várias camadas para o reconhecimento de um dígito presente em uma imagem.

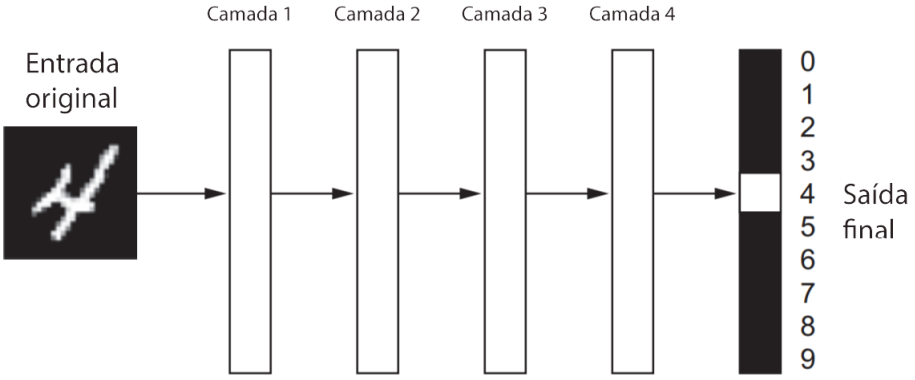


Figura 15 – Rede Neural profunda para reconhecimento de dígitos. Fonte: Adaptado de (CHOLLET, 2018)

Na Figura 16 a rede converte a imagem em representações diferentes da entrada original e, a cada nova camada, novas representações surgem, tais representações são baseadas na inserção da camada anterior. A cada nova representação existe um nível de detalhamento mais profundo, sobre o resultado final, do que as representações anteriores (CHOLLET, 2018).

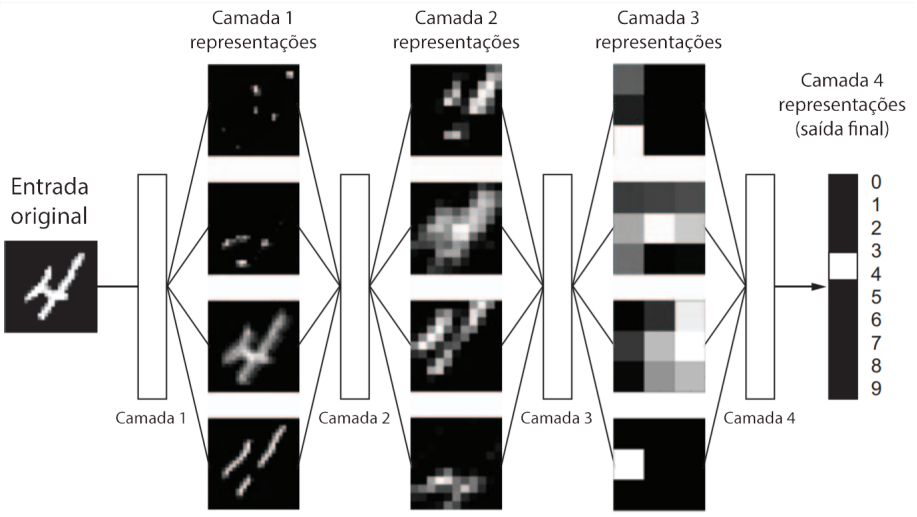


Figura 16 – Representações profundas aprendidas por um modelo de classificação de dígitos. Fonte: Adaptado de (CHOLLET, 2018)

2.8 Redes Neurais Convolucionais - CNN

Atualmente, a CNN é a técnica de DL mais popular no uso de visão computacional, destacando-se em aplicações que envolvem a extração de padrões em imagens, como no

reconhecimento de rostos e objetos, sem depender da extração manual desses recursos (MATHWORKS, 2020). Foi inicialmente desenvolvida por Y. Lecun et al. (LECUN et al., 1989) e classificada como uma rede *feedforward*, ou também chamadas de MLP (do inglês, *multilayer perceptron*) que tem por objetivo aprender os parâmetros que melhor descrevem a saída de um sistema. A visibilidade dessa técnica vem aumentando ano após ano, muito por conta dos resultados obtidos por ela.

Existem diversas arquiteturas de CNNs e suas variações, mas de modo geral, atualmente são compostas por 4 principais camadas internas, além das camadas de entrada e saída, sendo elas: I) Camada convolucional (ou de convolução), II) camada de *pooling*, III) Camada de não-linearidade e IV) camada totalmente conectada (KHAN et al., 2018). A Figura 17 mostra ilustra as camadas de uma CNN.

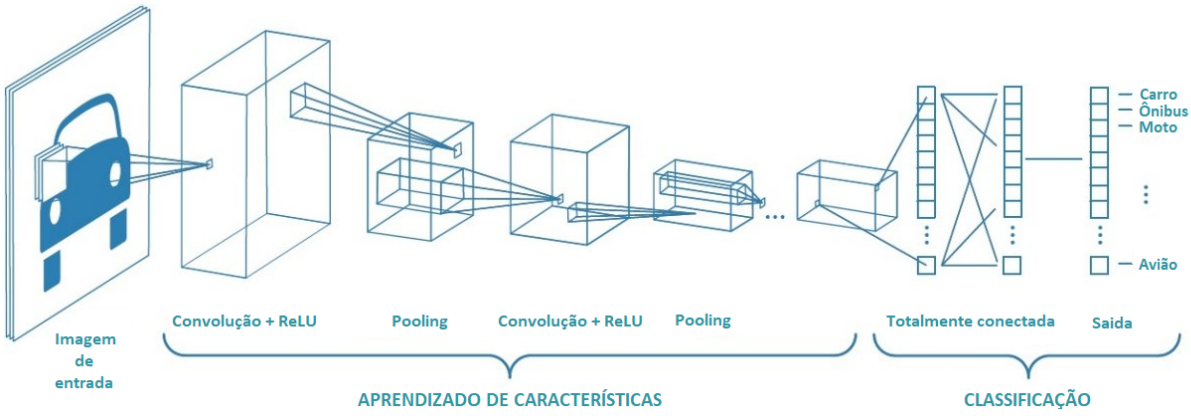


Figura 17 – Camadas típicas de uma CNN. Fonte: Adaptado de (MATHWORKS, 2020)

2.8.1 Camada Convolucional

A camada convolucional é a camada mais importante de uma CNN, já que é nessa camada que o sistema realiza a extração de características por meio de convoluções entre uma imagem de entrada e vários *Kernels* (ou filtro), onde cada *kernel* tem um determinado peso, o qual é inicialmente definido de maneira aleatória (também pode ser inicializado por diferentes outras abordagens (KHAN et al., 2018)) e a cada iteração no processo de aprendizagem esses pesos são ajustados. A camada convolucional é comumente aplicada a primeira camada de uma CNN.

A convolução de uma imagem bidimensional por um *kernel* é uma operação feita para a extração de caraterísticas da imagem, como a detecção de bordas (WANG et al., 2020), preservando as relações entre os pixeis da imagem, A saída dessa operação é uma matriz bidimensional chamada de Mapa de características (ou mapa de ativação).A Figura 18 mostra o processo realizado em uma entrada da camada convolucional. É importante ressaltar que na Figura 18 a operação realizada é a operação de correlação (e não de

convolução), embora a literatura de processamento de sinais define que essas operações são diferentes, na área de aprendizado de máquina, ambas as operações levam a convergência de pesos equivalentes, os dois termos são usados de maneira indistinta (KHAN et al., 2018).

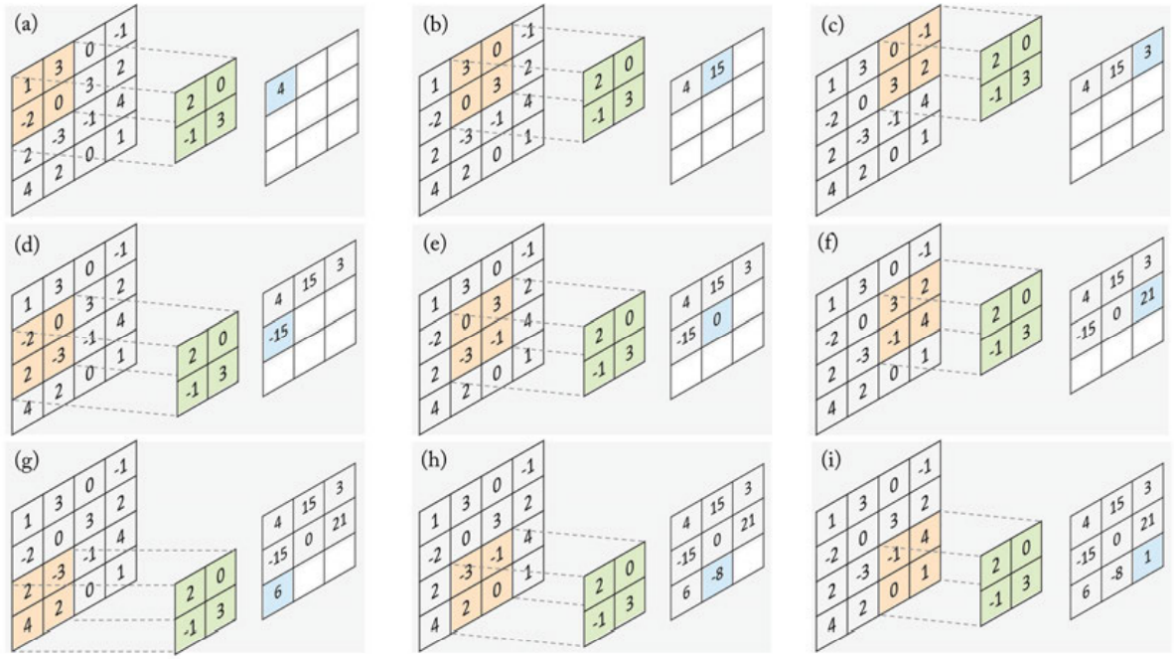


Figura 18 – Processo de correlação de uma matriz 4x4 por um *kernel* 2x2, *stride* de 1 e sem preenchimento. Fonte: (KHAN et al., 2018)

Na camada convolucional, parâmetros que precisam ser ajustados manualmente antes do início das convoluções são chamados de hiperparâmetros. Alguns desses parâmetros são apresentados a seguir.

- **Tamanho do *Kernel* -** O tamanho dos *kernels* que deslizam sobre as imagens bidimensionais interferem diretamente nos resultados de uma CNN, bem como os mapas de características obtidos a cada convolução. Por exemplo, é possível notar que *kernels* pequenos são capazes de extrair mais informações, obtendo um mapa de ativação de maior resolução, quando comparados a utilização de *kernels* maiores (WANG et al., 2020), entretanto, o custo computacional aumenta, já que a quantidade de informações processadas também aumenta.

- **Preenchimento -** Em aplicações como remoção de ruído de imagem, Super-resolução, ou segmentação, após a convolução é desejável manter a resolução dos mapas de características igual (ou até mesmo maior) a do mapa de entrada. isso possibilita redes de maiores profundidades, impedindo que a resolução dos mapas de recursos diminuam antes de obter os resultados desejados, o que pode apresentar melhorias nas rotulagens de saída de alta definição (KHAN et al., 2018). Essas características po-

dem ser alcançadas com o preenchimento (ou do inglês, *padding*), que nada mais é do que preencher as bordas dos mapas de características e/ou da imagem de entrada por determinados valores, sendo o preenchimento por zeros (ou do inglês, *zero-padding*) o mais comum dos preenchimentos. Outa vantagem do preenchimento é a preservação das bordas dos mapas de características, as quais podem ser importantes em algumas aplicações (WANG et al., 2020). Casos em que existe preenchimento são chamados de mesmo preenchimento (ou do inglês, *same padding*), já para os casos onde não existem preenchimento são chamados de preenchimento válido (ou do inglês, *valid padding*) (SAHA, 2018).

- ***Stride*** **-** A quantidade de *pixels* descolados ao deslizar o *kernel* (nas linhas e colunas) no processo de convolução é chamado de *stride*. Esse parâmetro também dita algumas características da CNN, já que para uma determinada resolução de imagem de entrada e um tamanho de *kernel* fixo, nota-se que a medida que o *stride* aumenta (para 2, por exemplo) obtem-se um mapa de características de menor resolução (quando comparado a um *stride* de 1, por exemplo) considerando mapas sem preenchimento.

2.8.2 Não-linearidade

É muito importante a utilização de uma função não linear após as camadas que definem os pesos para os parâmetros de uma rede (como nas camadas convolucionais e totalmente conectadas), já que isso possibilita uma rede neural aprender mapeamentos não lineares, ou seja, a saída não poderá ser uma combinação linear entre os pesos e uma entrada. Entre as diversas funções não lineares (algumas delas ilustradas na Figura 19) a Unidade Linear Retificadora ou ReLU (do inglês, *Rectifier Linear Unit*) destaca-se dentre as demais devido a seu baixo custo computacional e rápida convergência (KHAN et al., 2018). Basicamente, uma ReLU estabelece uma saída igual a 0 para entradas negativas e mantém o valor de entrada para valores positivos, como é descrito na equação 2.2. Essa análise é feita em todos os *pixels* dos mapas de características.

$$f_{ReLU}(x) = max(0, x) \tag{2.2}$$

2.8.3 Camada de agrupamento

A camada de agrupamento (do inglês, *pooling layer*) tem por objetivo diminuir a resolução da imagem, o que traz uma redução no custo computacional de implementação, já que a quantidade de parâmetros que a rede deve aprender será reduzido, sem perder informações de posição e translação, que são invariantes para mudanças razoáveis de escala (GOODFELLOW; BENGIO; COURVILLE, 2016). O processo de agrupamento

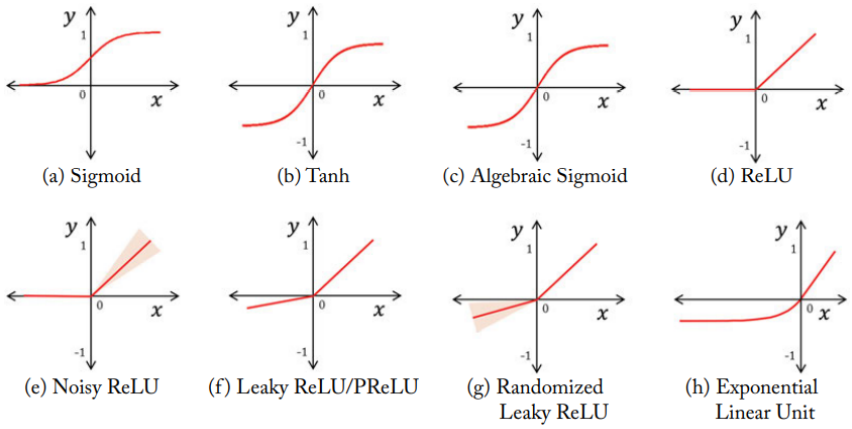


Figura 19 – Algumas funções não lineares comuns em arquiteturas de aprendizado profundo. Fonte:(KHAN et al., 2018)

é obtido ao deslizar uma janela de tamanho NxN a um passo definido no mapa de características após a n-ésima camada convolucional, de modo a selecionar (ou combinar) os valores obtidos do mapa dentro da janela estabelecida (WANG et al., 2020). No geral, a saída da camada de agrupamento é obtida a parir de três combinações: Seleção do maior valor presente na janela (agrupamento máximo, do inglês *max pooling*); Média dos valores presentes na janela (agrupamento médio, do inglês *average pooling*); Soma de todos os valores na janela (agrupamento de soma, do inglês *sum pooling*). é importante ressaltar que a camada de agrupamento diminui o tamanho do mapa de características, mas aumenta a dimensionalidade de acordo com a quantidade de *kernels* utilizados em cada iteração. A Figura 20 ilustra o processo de agrupamento máximo.

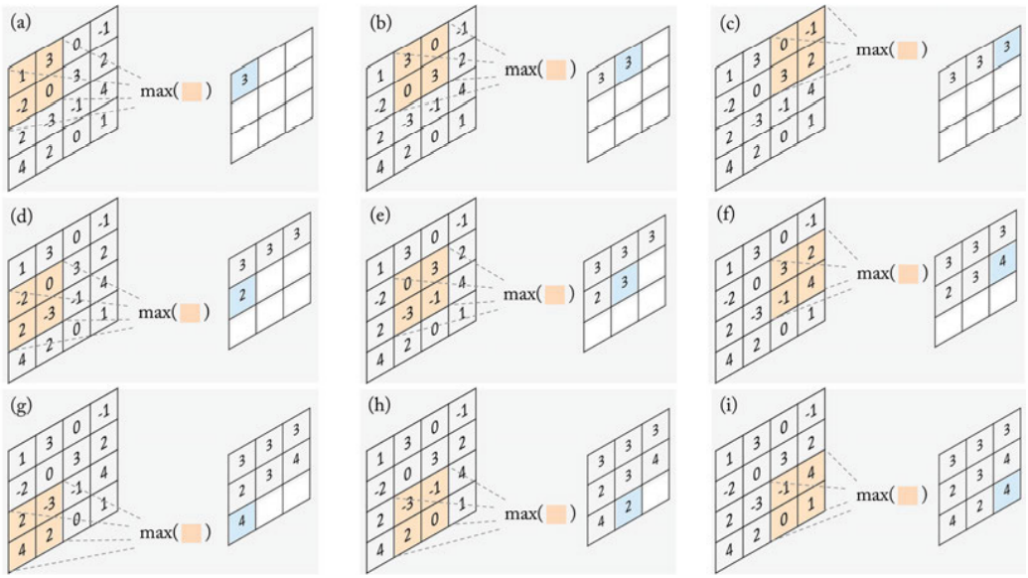


Figura 20 – Processo de agrupamento máximo com uma janela 2x2 e um passo de 1. Fonte:(KHAN et al., 2018)

Se a janela NxN e o *stride* (s) utilizados no processo de agrupamento são conhe-

cidos, é possível determinar o tamanho do mapa de características de saída, a partir das equações 2.3 (KHAN et al., 2018)

$$H' = \frac{H - N + s}{s}, W' = \frac{W - N + s}{s}, \tag{2.3}$$

em que H e W são respectivamente a altura e largura do mapa de recursos em que será aplicado o agrupamento, enquanto H' e W' representam a altura e largura do mapa após o agrupamento.

2.8.4 Camada de *upsample*

Em algumas aplicações, como a segmentação semântica, é necessário obter como saída mapas de tamanhos iguais ao de entrada, afim de ter uma máscara de segmentação, por exemplo. Para que essas informações perdidas durante os processos de convolução e agrupamento sejam reobtidas, são feitas operações de *upsampling* no mapa de características de saída, atuando como um gerador de formas (YIN; YAN; SHIN, 2019) (KHAN et al., 2018). Técnicas como interpolação bilinear, método do vizinho mais próximo, convolução transposta e *unpooling* são algumas das técnicas utilizadas nesse processo. O método de *unpooling*, para casos de agrupamento máximo (*max pooling*) por exemplo (Figura 21), é feito armazenando-se os índices de posições onde encontram-se os parâmetros selecionados pela janela de agrupamento, preenchendo todo os espaços restantes com zeros ou utilizando valores obtidos a partir de algum tipo de interpolação, como as citadas acima. A técnica de *unpooling* tem como vantagem a utilização de pouco memória, já que é necessário armazenar apenas o índice de posição de um agrupamento máximo, além de identificar posição e preenchimento de parâmetros (*max pooling*) (YIN; YAN; SHIN, 2019).

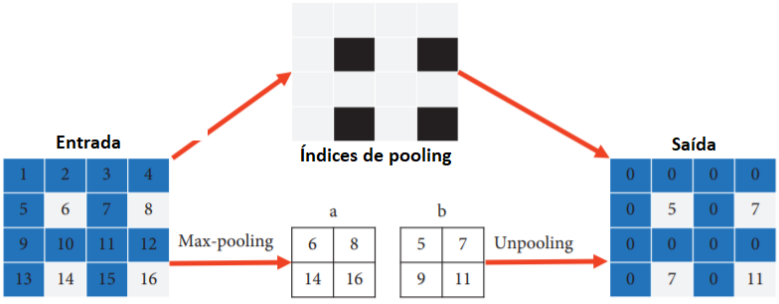


Figura 21 – Exemplo de um processo *upsample* utilizando o método de *unpooling*. Fonte: Adaptado de (YIN; YAN; SHIN, 2019)

2.8.5 Camada totalmente conectada

A camada totalmente conectada é geralmente implementada ao final da última camada de convolução/agrupamento, onde inicialmente realiza-se o achatamento dos mapas de características de saída, de modo a transforma-los em vetores coluna, por exemplo, para um tensor de saída 3x3x2 seria convertido para um vetor de tamanho 18 (WANG et al., 2020). Cada elemento desse vetor representa a probabilidade de um determinado recurso pertencer a uma determinada classe. Cada recurso tem um peso diferente para cada classe, esses pesos são calculados a partir do processo de *backpropagation*, os quais são readequados a cada iteração (MISSINGLINK.AI, 2020).

Conforme explicamos na Figura 22 a camada totalmente conectada garante que todos os nós (que contém os valores extraídos dos mapas de características) estarão conectados com os das próximas camadas e das camadas anteriores (ALBAWI; MOHAMED; AL-ZAWI, 2017). Até esse ponto, apenas foi feita a extração dos recursos e características das imagens. Para aplicações como classificação, é necessária uma função de ativação para normalizar as saídas da rede em um intervalo. No caso de problemas de classificação multi-classe, á função *softmax* é a função mais utilizada para essa situação (WANG et al., 2020; TIAN; FU, 2020). Essa função trata as saídas da camada totalmente conectada como a probabilidade de cada uma delas pertencer a uma classe **K** específica, a função está limitada entre o intervalo [0.1] (JOGIN et al., 2018). A função *softmax* é descrita pela equação 2.4,

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=0}^{K} e^{z_j}}. \tag{2.4}$$

Já para aplicações de classificação binária, a função sigmoide é bastante utilizada. Essa função também retorna um valor no intervalo [0,1], tendo como entrada um número real (KHAN et al., 2018). A função sigmoide é definida pela equação 2.5.

$$f_{sigmoid}(x) = \frac{1}{1 + e^{-x}}. \tag{2.5}$$

2.8.6 *Dropout*

O *dropout* ou abandono (tradução direta) é uma técnica de regularização de redes neurais criado por (SRIVASTAVA et al., 2014) que tende a evitar o *overfitting* do modelo, de modo a diminuir a quantidade de parâmetros que uma rede deve aprender durante a etapa de treinamento, o que reduz consideravelmente a complexidade da rede. Essa técnica consiste em, durante o processo de treinamento, zerar neurônios e suas conexões de maneira aleatória, de modo a definir uma probabilidade para que o *dropout* ocorra. Uma ilustração dessa técnica pode ser vista na Figura 23.

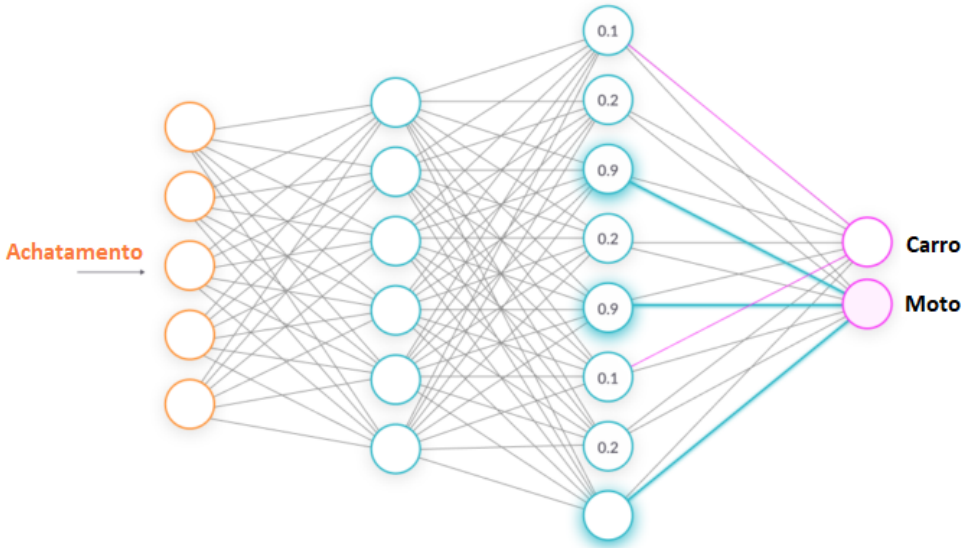
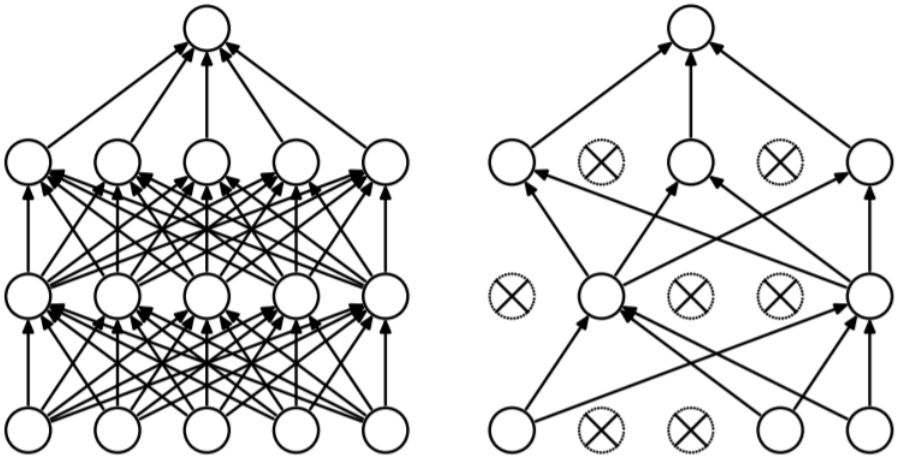


Figura 22 – Camada totalmente conectada com função *softmax*. Fonte:(MISSINGLINK.AI, 2020)



(a) Conexão padrão de uma rede neural (b) *Dropout* aplicado a rede neural

Figura 23 – Fonte: Adaptado de (SRIVASTAVA et al., 2014)

Essa técnica faz com que, ao desabilitar alguns neurônios, aqueles que continuam habilitados não terão mais ligação com aqueles que foram "desligados", isso faz com que a rede de neurônios remanescentes seja forçada a adquirir características mais robustas do problema em questão(VERGARA, 2018).

2.9 Arquiteturas de Redes Neurais Convolucionais

Com o passar dos tempos, várias arquiteturas de CNNs foram reformuladas e desenvolvidas, acarretando em avanços significativos para a área de aprendizado profundo.

Diversas arquiteturas famosas na literatura foram desenvolvidas para atuarem em escopos bem definidos, embora atualmente diversas dessas arquiteturas são utilizadas em várias outras aplicações e problemas, o que mostra a versatilidade da CNN.

Os avanços nessa área não param de acontecer. Competições mundiais de visão computacional utilizando tecnologias inovadoras para a classificação, detecção e segmentação de objetos podem ser um dos motivos para esse progresso. Desafios como o ImageNet Desafio de Reconhecimento Visual em Grande Escala ou ILSVRC (do inglês *Large Scale Visual Recognition Challenge*) e o desafio ISIC (do inglês, *Skin Lesion Analysis Towards Melanoma Detection*), fornecem os dados (bando de dados ou *datasets*) necessários para realização de treinamento, validação e testes dos modelos criados, o que fomenta ainda a aplicação desse tipo de tecnologia. Algumas das mais usuais e atuais arquiteturas são apresentadas a baixo, sendo elas: LeNet, VGG, ResNet, DenseNet e U-Net.

2.9.1 LeNet

O trabalho realizado por Lecun el al. (LECUN et al., 1998) para a classificação de caracteres manuscritos, implementou um arquitetura chamada de LeNet-5, essa arquitetura é das mais antigas CNNs existentes. A leNet-5 original utiliza uma imagem de entrada de tamanho 32x32 e é composta inicialmente por uma camada convolucional seguida de uma camada de agrupamento médio (termo em inglês *avarege pooling*), a arquitetura geral tem três camadas convolucionais com *kernels* 5x5 e duas camadas de agrupamento com janelas 2x2, obtendo-se mapas de características de dimensões 5x5x16 na quarta camada de agrupamento, após essa camada existem duas camadas totalmente conectadas (camadas 5 e 6) com 120 pesos e 84 pesos, respectivamente, em que são realizadas convoluções 1x1 entre todos os 400 parâmetro obtidos pelo achatamento dos mapas obtidos na quarta camada. por fim, é aplicada a função *softmax* para a classificação de dígitos de 0 a 9. A arquitetura LeNet-5 pode ser observada na Figura 24.

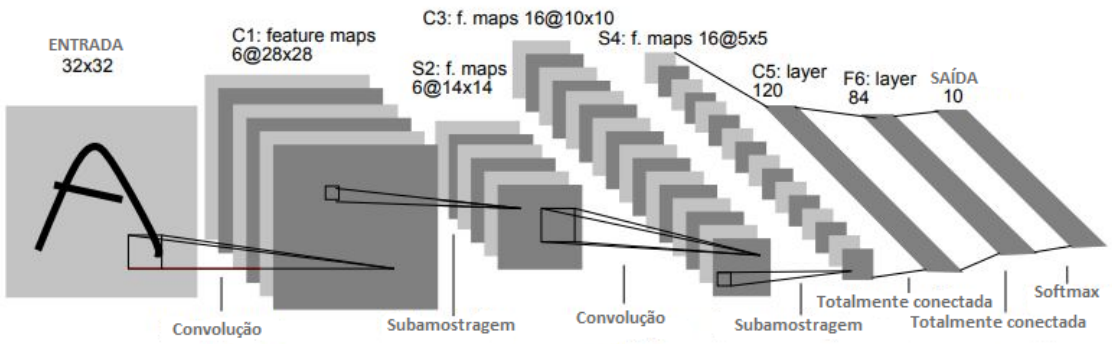


Figura 24 – Exemplo de uma arquitetura LeNet. Fonte: Adaptado de (LECUN et al., 1998)

2.9.2 VGG

A arquitetura VGG foi introduzida por Simonyan et al. (SIMONYAN; ZISSERMAN, 2014) e ficou conhecida por utilizar *kernels* pequenos (comparados a uma LeNet que usa *kernels* 5x5, por exemplo) nas camadas convolucionais. É composta por uma imagem de entrada de tamanho 224x224 seguida por duas camadas de convolução que utilizam *kernels* 3x3 com passo de 1 e *padding* para que as dimensões entre a primeira e segunda camada continue a mesma, obtendo-se campos receptivos efetivos de 5x5 seguidos de ReLU e *max pooling* com uma janela 2x2 e *stride* de 2, em 3 camadas são obtidos campos receptivos 7x7, devido a utilização de 3 convoluções. E por fim, são adicionadas três camadas totalmente conectadas, as duas primeiras com 4096 nós e a ultima 1000, por conta das 1000 classes que foram definidas para classificação.

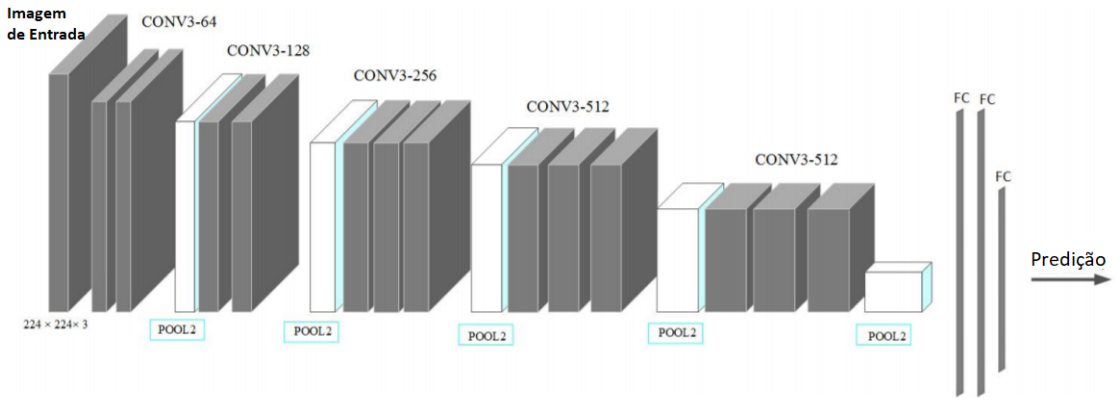


Figura 25 – Arquitetura VGGNet. Fonte: Adaptado de (MUHAMMAD et al., 2018)

Os motivos pelos quais são feitas duas convoluções 3x3 ao invés de apenas uma única convolução, são devido a diminuição de parâmetros que devem ser aprendidos, permitindo a rede convergir mais rapidamente, e ao aumento da quantidade de unidades não lineares, o que torna a função de decisão mais especifica (SIMONYAN; ZISSERMAN, 2014).

2.9.3 ResNet

Embora seja comum pensar que quanto maior a profundidade de uma CNN melhor será o resultado, ao tornarmos uma CNN muito profunda adicionando mais camadas empilhadas (em uma VGG, por exemplo), o problema de *vanishing gradient* aparece na etapa de treinamento, devido aos gradientes diminuírem exponencialmente a medida que a profundidade aumenta (GHAHREMANI; DROPPO; SELTZER, 2016). A arquitetura de Rede Residual ou ResNet (do inglês, *Residual Network*) Introduzida por Kaiming He et al. (HE et al., 2016) foi construída para solucionar esse problema.

Para a solução da *vaninshing gradient*, a ResNet usa blocos residuais (Figura 26), ao invés de aprender de modo que cada camada residual tem como entrada uma imagem/tensor de entrada que passa por convoluções e ReLU como em uma camada convolucional típica, mas a saída do bloco residual é a soma das camadas não-lineares empilhadas (convolução + ReLU) com a entrada do bloco residual, matematicamente $H(x) = F(x) + x$, em que $F(x)$ corresponde ao mapeamento típico de uma CNN e x a função identidade (saída igual a entrada). Um modelo de ResNet com 34 camadas pode ser vista na Figura 27.

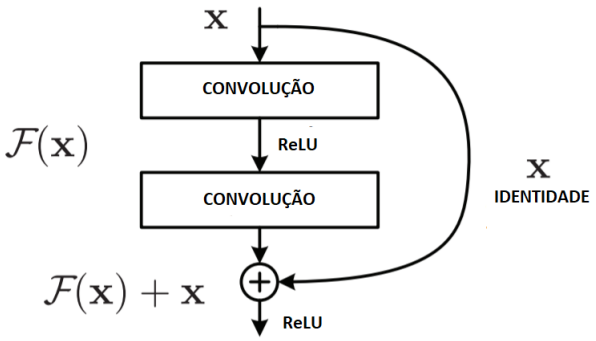


Figura 26 – Bloco residual de uma ResNet. Fonte: Adaptado de (HE et al., 2016)

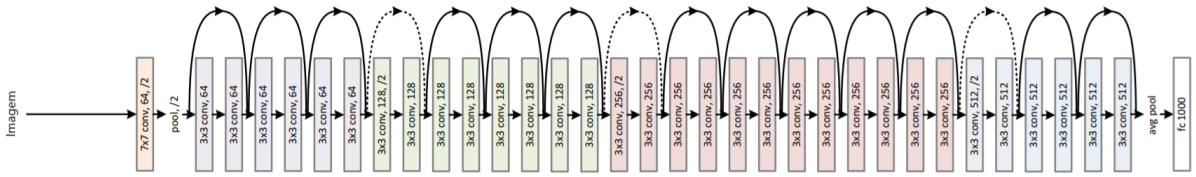


Figura 27 – Arquitetura ResNet com 34 camadas residuais. Fonte: Adaptado de (HE et al., 2016)

2.9.4 DenseNet

A rede convolucional densamente conectada (DenseNet) foi Idealizada por Gao Huang et al. (HUANG et al., 2017) com o intuito de aumentar ainda mais a profundidade das redes, de modo a evitar o *vanishing gradient*, criando canais para que as informações fluam entre as camadas, assim como a ResNet. Diferentemente da ResNet, a DenseNet realiza os *skips conections* de modo a ramificar a saída de uma camada para todas as camadas subsequentes, o que facilita a distribuição de informações em todas as direções, além de concatenar a saída de cada camada com as ramificações que ali estão presentes, ao contrário da soma que acontece na ResNet.

Uma DenseNet é composta por blocos densos e camadas de transição. Os blocos densos correspondem a várias camadas densas intercaladas por blocos que realizam a

convolução, *bacth normalization* e a ReLU. Cada camada densa representa a convolução de *kernels* 1x1 para extração de recursos, seguida de outra convolução com *kernels* 3x3 para diminuir a quantidade de canais para um valor k, chamado de taxa de crescimento (ou seja, a taxa de crescimento é a quantidade de canais de saída de uma camada densa). Já a camada de transição é responsável por diminuir as dimensões dos mapas de características de saída dos blocos densos, que corresponde a uma convolução 1x1 para diminuir o número de canais pela meta, seguida por um bloco de *average pooling* com uma janela 2x2 e *stride* de 2 (HUANG et al., 2017). As figuras 28 e 29 representam a arquitetura DenseNet.

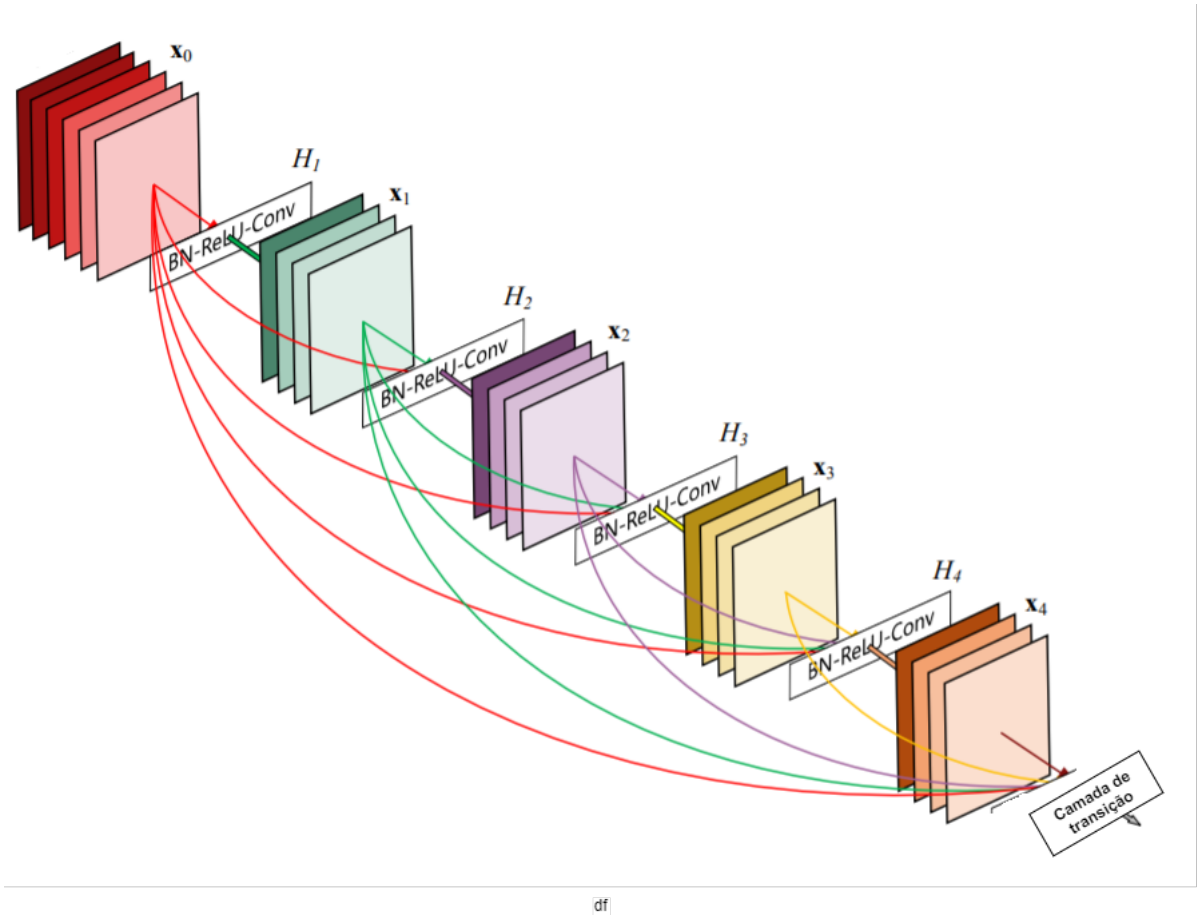


Figura 28 – Bloco denso de 5 camadas e taxa de crescimento $k = 4$, em que $x0, x1, x2$ e $x3$ são camadas densas. Fonte: Adaptado de (HUANG et al., 2017)

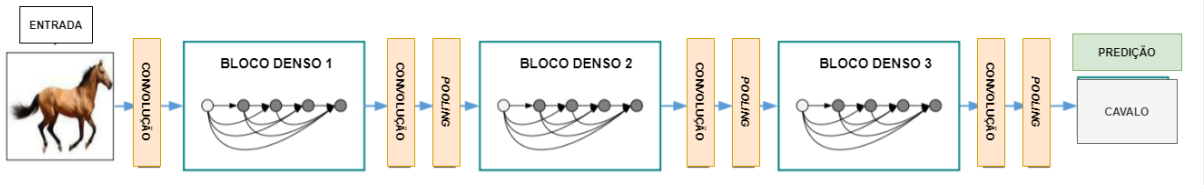


Figura 29 – Arquitetura DenseNet. Fonte: Adaptado de (HUANG et al., 2017)

2.9.5 U-Net

A U-Net é uma rede totalmente convolucional, mais comumente chamada de FCN (do inglês *Fully Convolutional Network*). É um tipo de CNN onde as camadas totalmente conectadas são substituídas por camadas convolucionais, conectadas apenas localmente (camadas de convolução, agrupamento e *upsample*) (SHELHAMER; LONG; DARRELL, 2017), artifício que torna possível obter como saída imagens de diversos tamanhos (como o mesmo tamanho das imagens de entrada, por exemplo), o que torna esse tipo de arquitetura muito útil em aplicações como segmentação, já que é possível rotular cada *pixel* para uma classe **K** com informações de localização espacial (DONG et al., 2019).

A U-Net foi criada e implementada por Ronneberger el al. (RONNEBERGER; FISCHER; BROX, 2015) com o prepósito de realizar a segmentação de imagens biomédicas. É compostas de duas parte, o caminho de contração (também chamado de *enconder*) responsável por extrair as características da imagem através das camadas típicas de uma CNN, e o caminho expansivo (ou *decoder*) onde camadas de *upsamples* (comumente chamadas de deconvoluções) são aplicadas, permitindo obter as posições dos *pixels* classificados, no caso da U-Net, o *decoder* tem a mesma quantidade de camadas que o *encoder*, ou seja, a quantidade de camadas de agrupamento e *upsample* são iguais, dando o formato de "U" a rede. é possível visualizar a arquitetura U-Net na Figura 30.

O *encoder* consiste em sequências de duas convoluções 3x3, seguida por uma ReLU e *max pooling* com uma janela 2x2 e passo de 2, onde a cada camada de convolução, dobra-se o numero de canais (profundidade) dos mapas de recursos. Já no *decoder*, são realizadas *upsamples*, dividindo por dois o número de canais do mapa de recursos presentes uma camada antes, em seguida é realizada a concatenação do mapa de saída dessa camada com o respectivo mapa (aquele que tem as mesmas dimensões) presente no caminho de *encoder* (olhar Figura 31) e por fim são realizadas duas convoluções 3x3 seguida de ReLU, permitindo identificar o "onde" e o "o que" da imagem (RONNEBERGER; FISCHER; BROX, 2015).

2.10 Métricas de Desempenho

As métricas de desempenho são índices usados para qualificar uma rede. Essas medidas podem ser obtidas a partir da matriz de confusão, que é uma ferramenta utilizada para descrever o desempenho de um classificador em um conjunto de dados de teste, onde sabe-se a verdade fundamental. No contexto de segmentação semântica de imagens, a verdade fundamental seria as máscaras de segmentação obtidas a partir do traçado manual (feito por um humano) dos limites de cada classe.

Essa análise é feita de modo que TP (verdadeiro positivo, do inglês *true positive*) representa o número de *pixels* que foram corretamente classificados como objeto alvo, TN

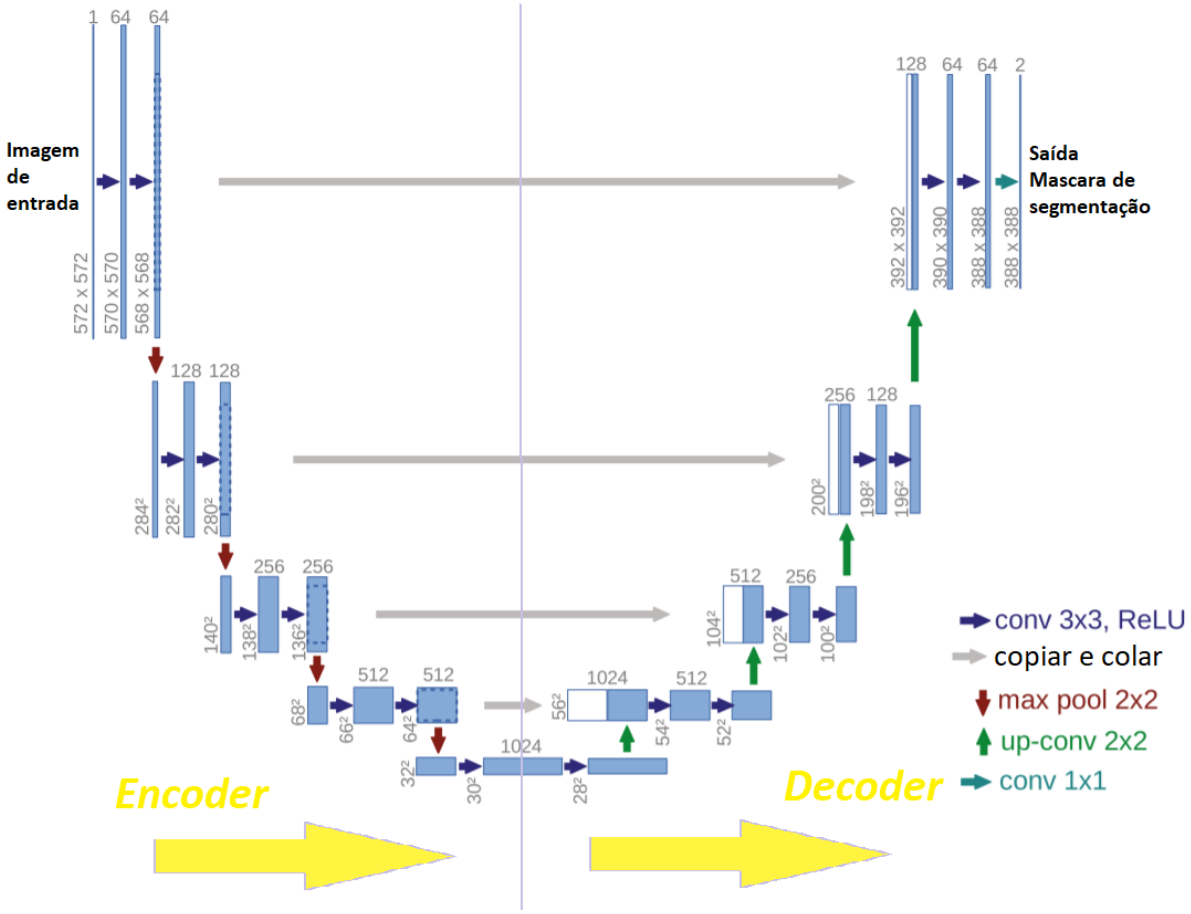


Figura 30 – Arquitetura U-Net. Fonte: Adaptado de (RONNEBERGER; FISCHER; BROX, 2015)

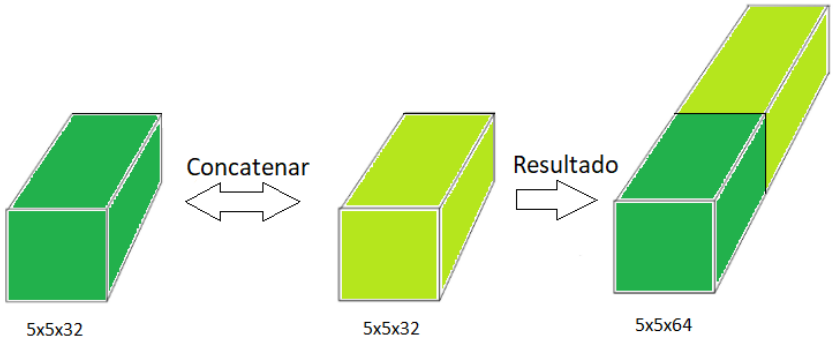


Figura 31 – Ilustração do processo de concatenar mapas de características, assim como ocorre em uma U-Net.

(verdadeiro negativo, do inglês *true negative*) o número de *pixels* que foram corretamente classificados como fundo, FP (falso positivo, do inglês *false positive*) o número de *pixels* que foram incorretamente classificados como obejto alvo e FN (falso negativo, do inglês *false negative*) o número de *pixels* que foram classificados incorretamente como fundo. A Figura 32 apresenta a matriz de confusão.

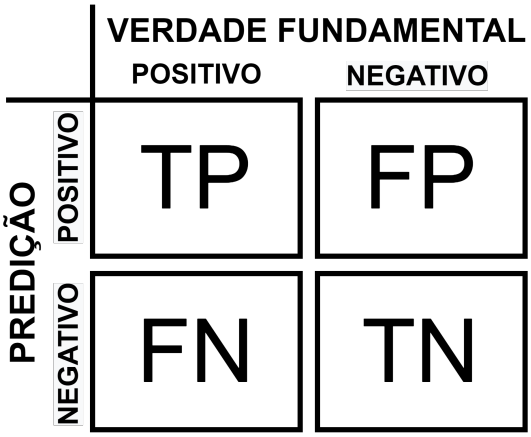


Figura 32 – Matriz de confusão.

Dentre as métricas mais utilizadas na qualificação de ANNs temos: Acurácia (AC), precisão (P), especificidade (EP) e sensibilidade (SE), coeficiente de Dice e índice de Jaccard (também chamado de IoU, do inglês *Intersection on Union*) (SILVA; PERES; BOSCARIOLI, 2016).

A acurácia (AC) representa a proporção total de classificações corretas

$$AC = \frac{TP + TN}{TP + TN + FP + FN}. \tag{2.6}$$

Já a precisão (P), é a proporção de valores positivos que foram corretamente classificados, em relação a todos os valores positivos

$$P = \frac{TP}{TP + FP}. \tag{2.7}$$

A especificidade (EP) pode ser descrita como a capacidade de prever corretamente amostras que não apresentam uma determinada doença

$$EP = \frac{TN}{TN + FP}. \tag{2.8}$$

Ao contrário da especificidade, a sensibilidade (SE) descreve a capacidade de prever corretamente amostras que existe uma determinada doença

$$SE = \frac{TP}{TP + FN}. \tag{2.9}$$

O coeficiente de Dice e o índice Jaccard são muito semelhantes e muito utilizadas na literatura para descrever a qualidade de segmentações. Esses índices também podem ser descritos como notação de conjuntos, como pode ser observado abaixo, onde A é o conjunto de dados preditos e B o conjunto de dados de verdades fundamentais

$$Jaccard = \frac{TP}{TP + FP + FN}, \tag{2.10}$$

$$Jaccard = \frac{|A \cap B|}{|A \cup B|}, \tag{2.11}$$

$$Dice = \frac{2.TP}{2.TP + FP + FN}, \tag{2.12}$$

$$Dice = \frac{2.|A \cap B|}{|A \cup B|}. \tag{2.13}$$

2.11 Trabalhos Relacionados

Dentre os trabalhos encontrados sobre segmentação de imagens dermatoscópicas, foram selecionados dois que utilizam o mesmo banco de dados (ISIC2017) e se mostraram interessantes para comparação. Ambos estão detalhados a seguir.

2.11.1 Jahanifar et al. (2018)

O trabalho de Jahanifar et al. (JAHANIFAR et al., 2018) propôs o uso da aprendizagem por transferência em uma nova estrutura de segmentação utilizando redes já conhecidas para isso. O treinamento, validação e previsão do modelo foram implementados usando a biblioteca Keras com o *backend* Tensorflow. Uma das diferenças dos demais trabalhos encontrados foi o uso extensivo de técnicas de aumento de dados, onde, além das técnicas padrões, como rotação da imagem, inversões aleatórias e aplicação de ruídos, buscou-se imitar a variação de aparência em imagens dermatoscópicas, como por exemplo, criando mapas de cabelo (pelos artificiais) através da caixa de ferramentas HairSim1 no Matlab e multiplicando os mapas aleatoriamente à imagem de entrada para sumular a presença de pelos reais nas manchas a serem segmentadas. Suas redes foram testadas em 3 bancos diferentes, mas para fins de comparação, utilizaremos os resultados apenas do banco de dados referente ao ISIC *challenge* do ano 2017 (ISIC2017), por ser o mesmo utilizado neste trabalho.

Jahanifar et al. (JAHANIFAR et al., 2018) baseou suas redes de segmentação na arquitetura UNet - que descrevemos neste mesmo capítulo -, utilizando uma de suas variantes para aumentar o desempenho e sendo capaz de realizar testes com diferentes redes. As redes pré-treinadas utilizadas para tal tarefa substituindo o bloco de rede base da UNet foram a ResNet de 152 camadas (ResNet152), a DensNet de 169 camadas (DensNet169), a Xception e a Inception-ResNet v2 (ResNetV2). Independente da rede utilizada, o *pooling* máximo ou camadas de convolução foram utilizadas cinco vezes em toda a arquitetura de

redes de base. Além de arquiteturas do tipo UNet, também foi incorporada a rede DeepLabV3 para a segmentação da mancha, uma arquitetura de rede de convolução avançada que utiliza a arquitetura Xception como rede base. Ao final de seu caminho de codificação, ainda foi incorporado um esquema de *pooling* de pirâmide espacial para aplicar convoluções dilatadas com taxas diferentes. Além dos procedimentos mencionados, foi utilizado o método *ensemble* (Ensemble), que consiste em uma combinação das máscaras de saída para todas as redes treinadas pelo autor.

Dentre elas, a de melhor desempenho para este banco de dados (ISIC2017) foram a ResNetV2 e o Ensemble, se levado em conta os índices de *Jaccard* de 80,2% e 80,6%, respectivamente, sendo estes os melhores resultados encontrados na literatura. Os valores das demais métricas podem ser observados na Tabela 1.

Tabela 1 – Resultados das arquiteturas construídas por Jahanifar et al.

	Jaccard(%)	*Dice(%)*	*AC(%)*	*P(%)*	*EP(%)*	*SE(%)*
DenseNet169	77,8	85,8	94,3	-	96,3	86,8
ResNet152	79,4	87,1	94,1	-	96,1	87,7
Xception	79,8	87,4	94,4	-	97,1	86,4
ResNetV2	80,2	87,6	94,4	-	96,4	87,3
Ensemble	80,6	87,9	94,6	-	96,9	87,9

No Capítulo 4 utilizamos apenas as duas redes com melhores desempenho desse trabalho (ResNetV2 e Ensemble) para que seja foi a comparação com os resultados que obtivemos.

2.11.2 Sheng Chen et al (2018)

Um dos trabalhos mais citados na literatura encontrada foi o de Sheng Chen et al. (CHEN et al., 2018). A sua rede foi treinada usando descida gradiente estocástica (SGD, do inglês, *Stochastic Gradient Descent*) com um minilote de 16 imagens. Para tanto, foram utilizadas as redes FeatureNet, ClsNet e SegNet. Para evitar overfitting, durnate a fase de treinamento foi utilizado técnicas de aumento de dados em tempo real, incluindo cortar, ampliar, girar, inverter e adicionar ruído gaussiano. Já para a fase de teste, foi aplicado corte, zoom e inversão nas imagens.

A FeatureNet foi retirada da parte da rede residual profunda da ResNEt-101 para se extrair características genéricas das imagens dermatoscópicas para ambas as tarefas de segmentação e classificação. Primeiramente, as redes FeatureNet e ClsNet sem SegNet foram treinadas por cerca de 80 épocas a uma taxa de aprendizado de 0,00001. Após o treinamento, o peso obtido pela ClsNet foi utilizado para inicializar e treinar a SegNet com FeatureNet e ClsNet fixadas, isto por cerca de 150 épocas na taxa de aprendizado (LR) de 0,0001. Finalmente, o módulo de passagem de recursos entre a ClsNet e a SegNet foi adicionado. Toda a rede foi treinada em conjunto por cerca de 45 épocas com LR de

0,0001. O tamanho de entrada das imagens foi redimensionado para 233 × 233, porém foi observado que o tamanho de entrada teve pouco impacto no desempenho. Como resultados do treinamento descrito, o autor apresenta um índice Jaccard de 79,8 como mostrado na Tabela 2.

Tabela 2 – Resultados da arquitetura construída por Sheng Chen et al.

	Jaccard(%)	*Dice(%)*	*AC(%)*	*P(%)*	*EP(%)*	*SE(%)*
Chen et al.	78,7	86,8	94,4	-	-	-

No trabalho (CHEN et al., 2018), também foram realizados testes para classificação das manchas porém, os valores apresentados nesta tabela (Tabela 2) são apenas para a etapa de segmentação por se mostrarem mais relevantes para a comparação com os resultados que obtivemos.

3 Metodologia

3.1 Base de Dados

Para que seja possível treinar os modelos de CNNs para a segmentação de melanoma, é necessária uma grande quantidade de imagens desse tipo de lesão de pele. A utilização de bancos de dados públicos contendo imagens dermatoscópicas de manchas de pele é a melhor maneira de se obter as imagens necessárias para o treinamento, devido a facilidade de aquisição desses dados, já que são licenciados para aplicações de pesquisa. Os bancos de dados ISIC e PH^2 são bastante utilizados para essa finalidade na literatura, como nos trabalhos de Manu Goyal et al. (GOYAL et al., 2020) e Jane Lameski et al. (LAMESKI et al., 2019).

- **ISIC -** O ISIC *archive* foi criado para o fomento na melhoria de técnicas de diagnóstico de melanoma, é patrocinado pela *International Society for Digital Imaging of the Skin* (ISDIS) e contém a maior quantidade de imagens dermatoscópicas de lesões cutâneas publicamente disponíveis. Neste trabalho serão utilizadas as imagens disponíveis no ISIC *Challenge* 2017, desafio lançado em 2017 contendo 2000 imagens dermatoscópicas junto as suas respectivas máscaras de segmentação (limites do contorno da lesão) definidas por profissionais dermatologistas que serão utilizadas na etapa de validação e análise de desempenho do modelo. Das 2000 imagens, 374 são melanoma, 254 ceratose seborreica e as 1372 imagens restante são nevos benignos. As imagens disponibilizadas para treinamento têm resolução $1024x768$ *pixels* com 8 *bits* de representação no espaço de cores RGB (ISIC, 2020).

- **PH^2 -** Já o PH^2 *database* contem um acervo de 200 imagens dermatoscópicas de lesões melanocíticas, incluindo 80 nevos comuns, 80 nevos atípicos e 40 melanomas, em que a segmentação realizada por dermatologistas também é fornecida. Trantam-se de imagens RGB de 8 *bits* com resolução de $768x560$ *pixels* (MENDONÇA et al., 2013).

Para a avaliação do desempenho de cada arquitetura utilizada, o banco de dados ISIC foi subdivido em treino, teste e validação com uma proporção de 70% para treino (1400 imagens), 25% para teste (500 imagens) e 5% (100 imagens) para validação. Essa subdivisão foi feita 3 vezes de forma aleatória, gerando 3 *datasets* diferentes $ISIC_a$, $ISIC_b$ e $ISIC_c$, afim de adicionar mais robustez aos dados resultantes, além de permitir avaliar o comportamento das redes para diferentes *datasets* de treinamento. As imagens pertencentes aos *datasets* de treinamento foram redimensionadas para a resolução 320x320, levando

em consideração o custo computacional (quanto maior a resolução da imagem, maior o custo computacional) e o fato de que a arquitetura U-Net necessita que a resolução das imagens utilizadas sejam divisíveis por 32.

O *dataset* PH^2 por sua vez, também foi redimensionado e foi utilizado somente como banco de teste para o modelo treinado a partir do *dataset* ISIC, devido a sua quantidade baixa de imagens. O objetivo principal foi analisar o comportamento do modelo treinado a partir de dados externos aos usados no treinamento.

3.2 Aumento de dados

Mesmo com um número limitado de dados, ainda é possível ampliar a quantidade desses dados a partir da técnica de aumento de dados (do inglês, *data augmentation*), que consiste em aplicar transformações nas imagens e suas respectivas máscaras de segmentação.

O método utilizado neste trabalho consiste em aplicar transformações em todas as imagens presentes no *dataset* de treinamento, substituindo as imagens originais pelas imagens transformadas, sem o aumento na quantidade de imagens de treinamento. Tal aplicação aumenta a variação nas imagens dentro dos *datasets* utilizados e entre eles, para que torne mais seguros os resultados obtidos nos treinamentos.

As transformações aplicadas estão presentes na Tabela 3. É importante ressaltar que é possível que mais de uma transformação seja feita na mesma imagem, isso também é feito de maneira aleatória. A Figura 33 apresenta algumas das técnicas utilizadas para aumento de dados, levando em consideração a tarefa de segmentação, já que algumas transformações utilizadas (como as mudanças de saturação, por exemplo) não são recomendadas para tarefas de classificação, já que aspectos como a cor da mancha é uma característica levada em consideração para o diagnóstico do melanoma, por exemplo.

Tabela 3 – Transformações utilizadas no segundo método de aumento de dados.

Transformações
Rotações aleatórias de 0^o a 360^o
Zoom aleatório (positivo e negativo)
Adição de ruído gaussiano
Alterações aleatórias de brilho e contraste
Aumento de nitidez
Aplicação de filtros de desfoque
Mudanças aleatórias de perspectiva
Alterações aleatórias de saturação

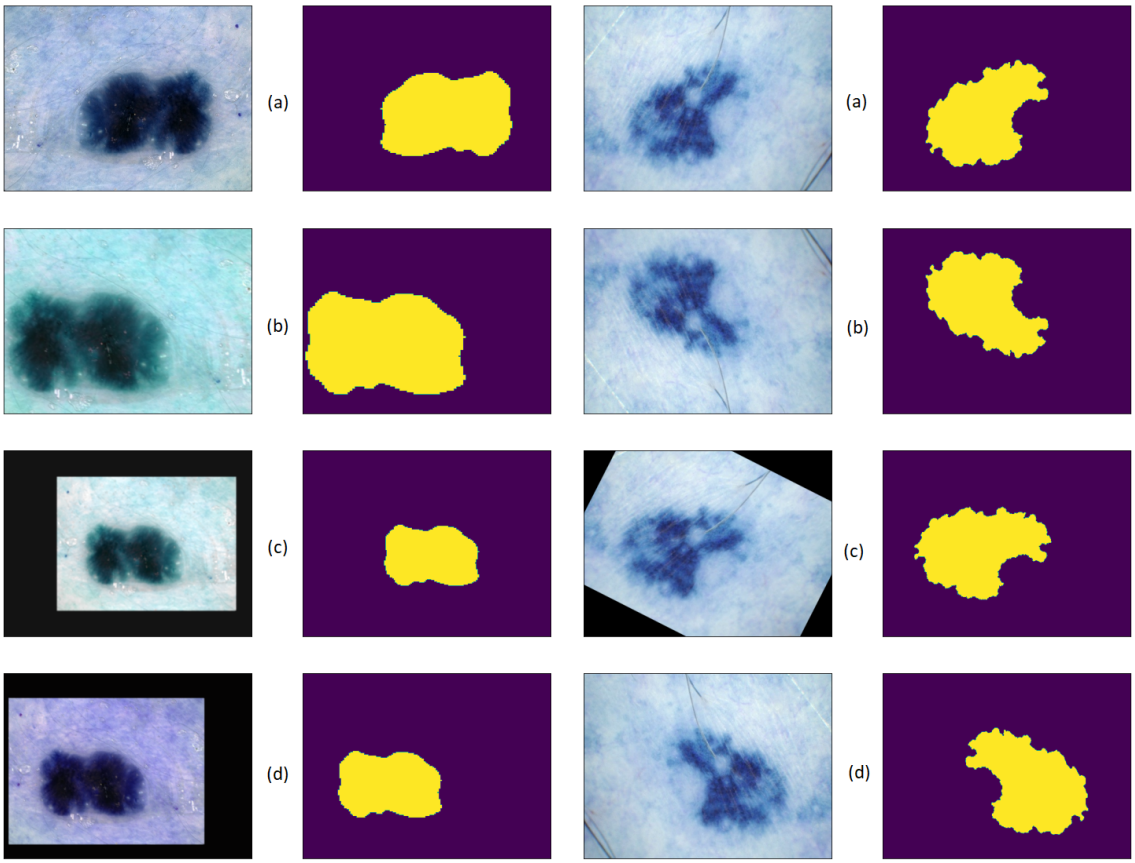


Figura 33 – Exemplos de algumas transformações aplicadas as imagens dermatoscópicas, e suas respectivas máscaras, para o aumento de dados. (a) imagem original, (b),(c), (d) transformações.

3.3 Arquiteturas Utilizadas

Para o processo de segmentação semântica de manchas de pele, foi utilizada a arquitetura U-Net, já que sua arquitetura nos permite alterar a estrutura do caminho de *encoder*, possibilitando a utilização de *backbones* (tradução direta do inglês, espinha dorsal) de CNNs tipicamente utilizadas em aplicações de classificação, para a tarefa de segmentação. Devido a flexibilidade desse tipo de arquitetura, serão feitas 3 combinações entre a U-Net e outras estruturas de CNNs atuando como , tanto no caminho de *encoder*, como no de *decoder*, afim de comparar o desempenho dessas arquiteturas. São elas:

- **U-Net + VGG-19:** A VGG com 19 camadas de peso (16 camadas convolucionais e 3 FC) como *encoder*, foi a configuração da arquitetura VGG que obteve os melhores resultados, dentre as 5 propostas pelos autores (SIMONYAN; ZISSERMAN, 2014).
- **U-Net + ResNet-50:** ResNet com 50 camadas residuais, é o intermédio entre os modelos ResNet-34 e ResNet-152, definido pela meio termo entre custo computacional e desempenho (HE et al., 2016).

- **U-Net + DenseNet-121:** DenseNet com 121 camadas de peso, é a configuração com a menor quantidade de camadas, embora não seja a configuração que apresenta os melhores resultados dentre as demais, foi escolhida levando em consideração as limitações no tempo de execução das plataformas que serão utilizadas (HUANG et al., 2017).

Em todas as três arquiteturas, a função utilizada para aplicar a não-linearidade entre as camadas da rede foi a ReLU, enquanto a técnica de *pooling* utilizada foi a *max pooling*, em contrapartida a técnica utilizada nas camadas de *upsample* foi a interpolação *Nearest Neighbors*.

3.4 Recursos Computacionais

Em relação ao projeto serão necessários recursos computacionais de hardware e software.

3.4.1 *Hardware*

O ambiente utilizado para a implementação deste trabalho foi o Google Colab. O Google *Colaboratory*, ou "*Colab*" é um serviço de nuvem gratuito que permite escrever e executar códigos em Python por meio do navegador. O Colab é um serviço hospedado de *notebook* que não requer configuração para uso, porém fornece recursos de computação usados em DL. Possui sua versão gratuita e sua versão paga (Google Colab Pro), cada versão possui suas peculiaridades em relação a limite de inatividade da seção, GPU, tempo de execução, memória entre outros. Essas particularidades podem ser observadas na Tabela 4 a seguir.

Tabela 4 – Comparação entre CPUs das máquinas virtuais

Parâmero	**Google Colab**	**Google Colab Pro**
Modelo GPU	K80	T4 ou P100
Tempo de execução	16 horas	24 horas
RAM disponível	12GB	25GB

Inicialmente o projeto foi desenvolvido no ambiente do Google Colab em sua versão gratuita, porém foram encontradas barreiras que atrapalhavam o desenvolvimento do projeto em tempo hábil. Uma das barreiras enfrentadas foi o tempo de execução das épocas no treino de cada rede, onde a média era de aproximadamente 2 minutos. Portanto um treino com 100 épocas demorava em torno de 3 horas para ser concluído.

A baixa disponibilidade da utilização do GPU também foi uma barreira encontrada na versão gratuita, pois nesta versão, após o consumo total do GPU, o mesmo tinha seu

fornecimento interrompido, tornando assim inviável continuar o treinamento da rede sem esse recurso devido ao aumento expressivo no tempo de execução de cada época. Por fim a inatividade da seção apresentava um tempo baixo, cerca de 40 minutos, onde após a detecção de inatividade o ambiente era desconectado e todo o processo feito anteriormente era perdido.

Devido as dificuldade apresentadas anteriormente houve a necessidade da utilização do ambiente em sua versão Pro. Com a utilização da versão Pro, o tempo de execução de cada época caiu para menos da metade, não houve mais problemas com uso total do GPU e nem inatividade de seção.

3.4.2 *Software*

Para a construção ou utilização das arquiteturas de DL, serão necessárias bibliotecas. As principais bibliotecas utilizadas no ramo de DL são o TensorFlow, Keras.

O TensorFlow é uma biblioteca de código aberto amplamente utilizada para aprendizado de máquina e suas aplicações. Possui diversos recursos de diferenciação automática que tornam o processo de definição de novas redes mais simples. As linguagens suportadas pelo TensorFlow são o python e C++. Oferece ainda suporte a vários *back-ends*, CPU ou GPU em *desktops*, servidores ou plataformas móveis.

O Keras é uma API em código aberto para redes neurais de alto nível. É escrita em Python e capaz de ser executada no TensorFlow e Theano - biblioteca Python usada para definir, otimizar e avaliar expressões matemáticas envolvendo matrizes multidimensionais. O Keras proporciona uma experiência mais rápida com redes neurais profundas que permite ir da ideia aos resultados da maneira mais rápida possível. Contudo o Keras não fornece a maioria dos modelos pré-treinados de última geração. A Tabela 5 apresenta características presentes em cada biblioteca (KHAN et al., 2018).

Tabela 5 – Comparação de bibliotecas usadas para Aprendizado Profundo

Software	**Criador**	**Plataforma**	**Linguagem**	**Interface**
TensorFlow	Google	Linux, Mac OS X, Windows	C++, Python	Python, C/ C++, Java, Go
Keras	Francois Chollet	Linux, Mac OS X, Windows	Python	Python

3.5 Implementação

3.5.1 U-Net

A arquitetura U-Net foi implementada em TensorFlow e Keras com a utilização de uma biblioteca desenvolvida por (YAKUBOVSKIY, 2019), que disponibiliza modelos de *backbones* como VGG, ResNet, DenseNet e vários outros, com pesos pré-treinados do *dataset* ILSVRC 2012 (do inglês, *ImageNet Large Scale Visual Recognition Challenge*) ImageNet, tornando o processo de treinamento mais rápido.

3.5.2 Hiperparâmetros

Os hiperparâmetros utilizados para a implementação das arquiteturas com seus respectivos *backbones* podem ser observados na Tabela 6, os mesmos foram ajustados a a partir de testes de modo a se obter o melhor modelo possível, sempre levando em consideração evitar o *overfitting*, variando-os um a um. As épocas divergiram os valores devido a constatação de estabilidade nas métricas a partir de certos valores inseridos, já o *batch* variou para a DenseNet-121 devido a mesma apresentar melhores resultados com o valor acima dos demais. O valor do *Batch* remete a quantidade de grupos em que os dados de treinamento foram divididos, em que para cada época apenas um desses grupos será utilizado, ou seja, para um $Batch = 8$, tem-se $12,5\%$ dos dados de treinamento por época.

Tabela 6 – Hiperparâmetros utilizados.

Hiperparâmetro	**DenseNet-121**	**ResNet-50**	**VGG-19**
Épocas	100	100	50
Batch	4	8	8
Taxa de Aprendizagem	0,0001	0,0001	0,0001

3.5.3 Função de Perda

Como dito no Seção 2.6.2, uma função de perda é utilizada como *feedback* para corrigir os pesos das camadas durante o treinamento, a fim de obter as predições mais próximas do real valor. Uma dessas funções de perda é a perda de entropia cruzada binária ou BCE (do inglês, *Binary Cross-Entropy*), que é uma generalização para casos de classificação binária da entropia cruzada ou CE (do inglês, *Cross-Entropy*). A entropia cruzada (CE) é representada pela equação 3.1

$$CE(p,t) = -\sum_{n} t_n \log(p_n), \qquad n \in [1,N], \tag{3.1}$$

onde t_n representa a verdade fundamental, que é o valor real da classe analisada, enquanto p_n é a respectiva predição normalizada (normalizada no intervalo [0,1] a partir de uma função de ativação, explicado mais a frente) e N o número de classes. Para o caso binário ($N = 2$), podemos expandir a equação 3.1, obtendo a equação 3.2,

$$CE = -t_1 \log(p_1) - t_2 \log(p_2). \tag{3.2}$$

Com base nisso, podemos assumir que $t_1 + t_2 = 1$, já que um *pixel* só pode pertencer a uma única classe, de maneira análoga temos que $p_1 + p_2 = 1$, isolando t_2 e s_2 e substituindo-os na equação 3.2, obtém-se então a perda de entropia cruzada binária (equações 3.3 e 3.4),

$$BCE = -\log(p_1), \quad \text{se} \quad t = 1, \tag{3.3}$$

$$BCE = -\log(1 - p_1), \quad \text{se} \quad t = 0. \tag{3.4}$$

outra função de perda interessante é a Perda Focal ou FL (do inglês, *Focal Loss*), a qual trata-se de uma perda CE multiplicada por um fator de modulação $(1 - p_n)^\gamma$ e pesos de ponderação α, em que γ é chamado de parâmetro de foco, como pode ser observado na equação 3.5,

$$FL(p, t) = -\sum_n \alpha(1 - p_n)^\gamma t_n \log(p_n), \qquad n \in [1, N], \tag{3.5}$$

tais parâmetros são adicionados com o objetivo de fazer com que a função de perda pondere as predições, de modo à reduzir o peso de predições bem classificadas e aumentar o peso de predições ainda incertas, fazendo com que a contribuição para a função de perda seja maior para predições ainda incertas em relação as predições bem classificadas, levando-se em consideração os parâmetros γ e $\alpha > 0$.

De maneira semelhante ao que foi feito pare obtermos a equação 3.3 a partir da equação 3.1, podemos definir a perda focal binária, para $t = 1$, como:

$$FL(p) = -\alpha(1 - p)^\gamma \log(p). \tag{3.6}$$

A FL e a CE são comparadas a partir do gráfico presente na Figura 34 (Lin et al., 2017), (KHAN et al., 2018).

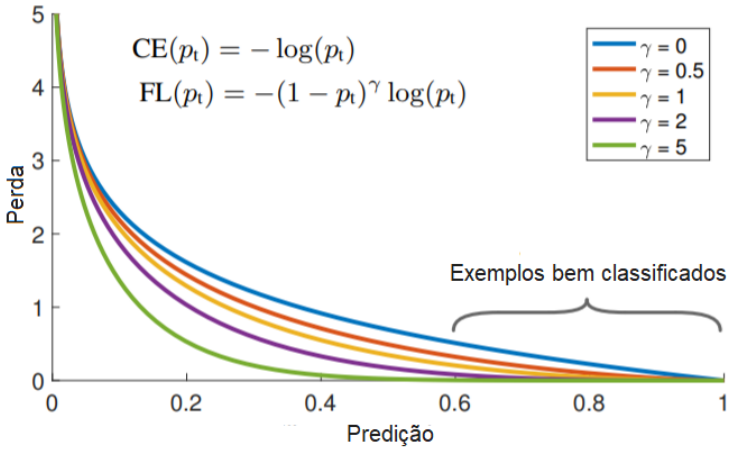


Figura 34 – Funções de perda CE e FL, para valores variados de γ e $\alpha = 1$. Fonte: Adaptado de (Lin et al., 2017)

A função de perda definida para a aplicação em questão, foi escolhida como sendo a perda focal (FL) com parâmetro de foco $\gamma = 2$, somada a outra função de perda, baseada no coeficiente de Dice, sendo ela (DL):

$$DL = 1 - Dice, \tag{3.7}$$

já que a mesma é amplamente utilizada em tarefas de segmentação (JAHANIFAR et al., 2019), pois trata-se da sobreposição de duas amostras. Outro motivo para essa função ser utilizada somada a FL, é para compensar a perda baixa em módulo da FL, devido ao parâmetro de foco γ que aumenta a perda de amostras mal classificadas em relação as amostras bem classificadas, mas diminui esses valores em modulo quando comparada a outros tipos de perda, como a CE. Logo a função de perda (L) utilizada é expressa pela equação 3.8.

$$L = (1 - Dice) + FL. \tag{3.8}$$

3.6 Métricas de Desempenho

A fim de comparar o desempenho das arquiteturas utilizadas de maneira objetiva, é necessário qualificar o desempenho de cada uma dessas arquiteturas. Métricas como o índice de Jaccard, coeficiente de Dice, acurácia, precisão, especificidade e sensibilidade são medidas que foram utilizadas para mensurar o desempenho de um modelo de segmentação a partir da matriz de confusão. A verdade fundamental, no caso da segmentação de melanoma, corresponde às máscaras de segmentações obtidas por dermatologistas e a

partir delas serão extraídos os valores correspondentes necessários para o cálculo de cada índice.

3.7 Procedimento de teste

A fim de padronizar os procedimentos de teste, foi criado um roteiro listando a ordem de cada etapa a ser seguida. Em cada um dos *datasets* utilizados, os procedimentos de teste foram feitos primeiramente com a presença do aumento de dados e posteriormente sem a utilização do aumento de dados, a fim de avaliar o comportamento do modelo. De forma semelhante, todo o procedimento citado abaixo foi feito para cada *backbone* utilizado. O procedimento de teste foi dividido em:

- **Escolha do banco de dados:** Nesta etapa a escolha do banco de dados seguiu a ordem de $ISIC_A, ISIC_B$ e $ISIC_C$. Cada banco de dados já estava subdividido em treino, teste e validação, novamente.
- **Inserção do *backbone*:** Neste estágio são seleciona-se o *backbone* desejado, estruturado na arquitetura U-Net.
- **Hiperparamentros:** Aqui foram feitas combinações dos hiperparametros (tamanho do *batch*, número de épocas e taxa de aprendizagem), a fim de obter o melhor modelo possível.
- **Treinamento e obtenção do modelo:** Nesta etapa a rede era treinada de acordo com os parâmetros passados anteriormente, tal como o banco de dados utilizado e ao final do treinamento o modelo era obtido. O acompanhamento das métricas de validação fez-se necessário para a identificação do comportamento da rede sem que fosse necessário terminar o treinamento, tornando possível perceber comportamentos como o *overffiting*.
- **Obtenção das métricas e máscaras:** Esta etapa é destinada à obtenção das métricas (definidas anteriormente) obtidas a partir do modelo após o treinamento referentes aos *datasets* de treino, validação e teste. Com relação as máscaras preditas pelos modelos treinados de cada arquitetura, utilizando-se o *dataset* de teste, em que o modelo recebe como entrada uma imagem presente nesse conjunto e realiza a predição da mesma, retornando uma imagem do tipo *float* tendo *pixels* com valores no intervalo $[0, 1]$, em que escolheu-se como limiar de binarização o meio termo $0,5$, logo, tudo maior que esse limiar foi transformado em 255 e abaixo desse valor agregou-se aos *pixels* o valor 0, transformando a máscara predita em uma imagem em tons de cinza. É importante ressaltar que a matriz de confusão, utilizada para obter as métricas apresentadas, foi obtida a partir das máscaras binarizadas.

- **Aplicação externa:** Etapa com o objetivo de verificar o desempenho do modelo quando aplicado ao *dataset* PH^2, o qual foi utilizado como *dataset* de teste, afim de obter as métricas de desempenho e as máscaras preditas pelo modelo.

4 Resultados e Discussões

Neste capitulo são apresentados os resultados obtidos seguindo-se os aspectos definidos na metodologia, bem como suas respectivas discussões. Esses resultados são expressos em formas de tabelas, gráficos e figuras, de modo a facilitar o entendimento ao leitor.

As Tabelas 7, 8 e 9 apresentam os resultados obtidos a partir do treinamento das arquiteturas propostas, levando em consideração a aplicação do método de aumento de dados (índices **(b)** das respectivas tabelas) descrito no capítulo anterior.

Tabela 7 – Métricas U-Net+DenseNet-121

(a) Sem aumento de dados

	$ISIC_A$	$ISIC_B$	$ISIC_C$
Jac (%)	81,69	80,50	78,95
Dice (%)	90,15	89,22	88.23
AC (%)	96,05	96,12	96,15
P (%)	95,86	95,15	96,50
EP (%)	98,80	98,98	99,35
SE (%)	84,50	83,98	81,27

(b) Com aumento de dados

	$ISIC_A$	$ISIC_B$	$ISIC_C$
Jac (%)	83,64	81,71	78,21
Dice (%)	91,09	89,94	87,77
AC (%)	96,57	96,40	95,91
P (%)	95.75	96,52	94,57
EP (%)	99,02	99,28	98,97
SE (%)	86,86	84,19	81,89

Tabela 8 – Métricas U-Net+ResNet-50

(a) Sem aumento de dados

	$ISIC_A$	$ISIC_B$	$ISIC_C$
Jac (%)	80,30	79,91	78,01
Dice (%)	89,07	88,83	87,65
AC (%)	95,87	95,96	95,91
P (%)	95,53	94,41	95,52
EP (%)	99,01	98,82	99,17
SE (%)	83,73	83,87	80,97

(b) Com aumento de dados

	$ISIC_A$	$ISIC_B$	$ISIC_C$
Jac (%)	82,63	82,26	80,94
Dice (%)	90,48	90,27	89,46
AC (%)	96,34	96,46	96,46
P (%)	95,16	95,15	95,68
EP (%)	98,89	98,96	99,17
SE (%)	86,24	85,85	84,03

Tabela 9 – Métricas U-Net+VGG-19

(a) Sem aumento de dados

	$ISIC_A$	$ISIC_B$	$ISIC_C$
Jac (%)	79,75	79,10	77,84
Dice (%)	88,73	88,33	87,56
AC (%)	95,51	95,77	95,76
P (%)	89,85	93,59	92,27
EP (%)	97,50	98,64	98,48
SE (%)	87,64	83,63	83,30

(b) Com aumento de dados

	$ISIC_A$	$ISIC_B$	$ISIC_C$
Jac (%)	80,24	79,75	79,14
Dice (%)	91,38	88,73	88,35
AC (%)	94,57	95,86	95,99
P (%)	93,71	92,43	92,07
EP (%)	97,15	98,34	98,40
SE (%)	89,16	85,32	84,92

A Tabela 10 apresenta a média das métricas obtidas utilizando o *dataset* ISIC, de modo que para cada métrica a média é dada por $(ISIC_A + ISIC_B + ISIC_C)/3$. Como

pode ser visto na mesma tabela, a arquitetura U-Net + ResNet-50 foi a que apresentou o melhor desempenho entre as estudadas se levado em consideração o média do índice Jaccard. Apesar isso, o melhor modelo obtido foi apresentado pela arquitetura DenseNet-121 quando treinada com o *dataset* $ISIC_A$ utilizando o aumento de dados. Também é possível observar que a aplicação da técnica de aumento de dados melhorou os resultados em relação aos modelos que não utilizaram tal técnica.

Tabela 10 – Comparação entre a média das métricas do *dataset* ISIC

(a) Sem aumento de dados

	Jaccard(%)	*Dice(%)*	*AC(%)*	*P(%)*	*EP(%)*	*SE(%)*
DenseNet-121	80.38	89.20	96.10	95.83	99.04	83.25
ResNet-50	79.40	88.51	95.91	95.15	99.00	82.76
VGG-19	78.89	88.20	95.68	91.90	98.20	84.85

(b) Com aumento de dados

	Jaccard(%)	*Dice(%)*	*AC(%)*	*P(%)*	*EP(%)*	*SE(%)*
DenseNet-121	81.19	92.93	96.29	95.61	99.09	84.31
ResNet-50	81.94	90.07	96.42	95.33	99.01	85.37
VGG-19	81.00	89.48	95.47	92.73	97.96	86.46

A Tabela 11 apresenta os resultados das métricas obtidas para as predições feitas pelo modelo, utilizando o *dataset* PH^2. Os modelos utilizados foram aqueles treinados com as imagens presentes no *dataset* $ISIC_A$ com aumento de dados, que foi o que apresentou as melhores métricas para as 3 arquiteturas estudadas.

Tabela 11 – Métricas obtidas para o *dataset* PH^2

	Jaccard(%)	*Dice(%)*	*AC(%)*	*P(%)*	*EP(%)*	*SE(%)*
DenseNet-121	84.61	91.66	94.78	94.57	97.56	88.93
ResNet-50	84.18	91.41	94.64	94.66	97.62	88.38
VGG-19	83.66	91.10	94.43	94.20	97.41	88.20

A partir das métricas obtidas na utilização do *dataset* PH^2 para testar os modelos (Tabela 11), é possível revalidar o modelo e suas configurações, já que não existe sinais de *overfitting*. Como pode ser observado, as métricas obtidas para esse *dataset* são melhores se comparadas as métricas obtidas utilizando o *dataset* ISIC como teste, isso se dá pelo fato do *dataset* PH^2 conter uma quantidade inferior de imagens, além de que as imagens presentes nesse conjunto são visualmente mais agradáveis, contendo menos pelos, por exemplo.

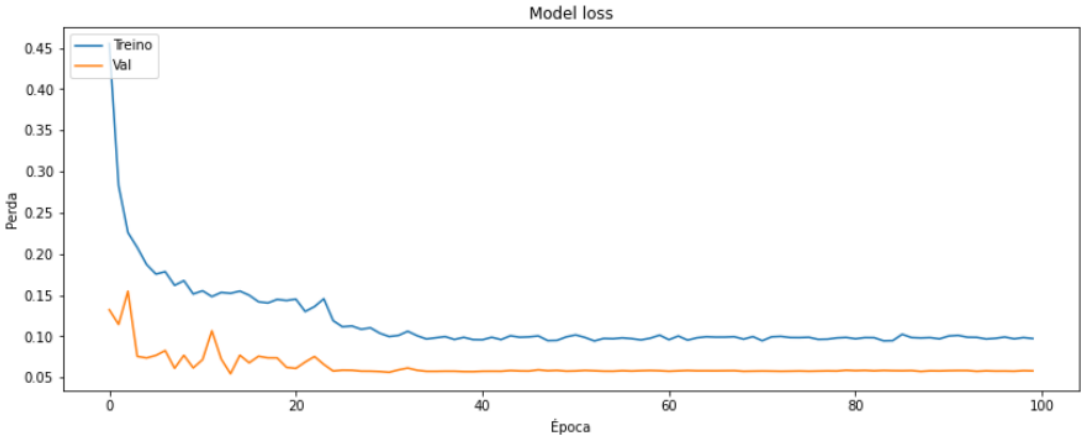
Com base nas métricas apresentadas nas Tabelas 10 e 11, nota-se que os modelos treinados são modelos de especificidade muito alta, um dos motivos disso se dá devido as manchas presentes nos *datasets* apresentarem lesões que não são dissipadas em outras regiões da mesma imagem, tendo uma única região por imagem, na maioria dos casos.

Outro fator para essa alta especificidade está relacionada a proporção das manchas em relação as pele saudável (ou fundo), já que esses *datasets* apresentam manchas menores em proporção ao fundo, tornando o modelo mais acertivo na classificação do fundo.

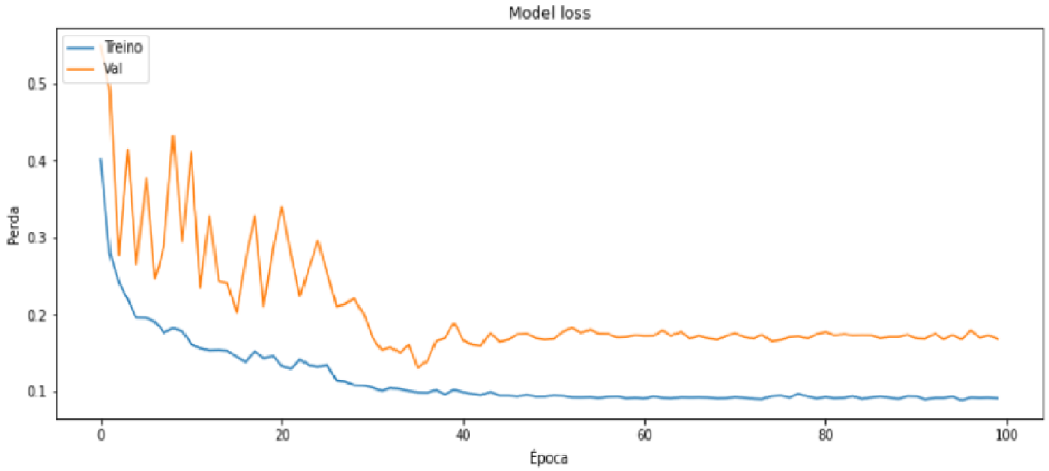
Do ponto de vista clínico, modelos de alta especificidade são uma ótima ferramenta. No caso de melanoma, por exemplo, se o modelo identificar uma região como negativa (sem melanoma), essa região deve ser realmente negativa, pois caso essa região fosse realmente positiva (com melanoma), o modelo estaria desconsiderando partes da mancha, podendo provocar sérios danos a pacientes expostos a essa má predição, como por exemplo mostrando uma gravidade da situação menor que a real.

Abaixo, a Figura 35 apresenta o gráfico da variação da função de perda durante as épocas do treinamento da U-Net, juntamente com cada *backbone* para o *dataset* $ISIC_A$ com aumento de dados. Como pode ser observado na Figura 35(a) e 35(c), a perda no *dataset* de validação foi menor que a perda no *dataset* de treinamento em alguns pontos (e até durante todo o treinamento), o que é um fato plausível se for levado em consideração a quantidade baixa de imagens no *dataset* de validação em relação ao *dataset* de treinamento.

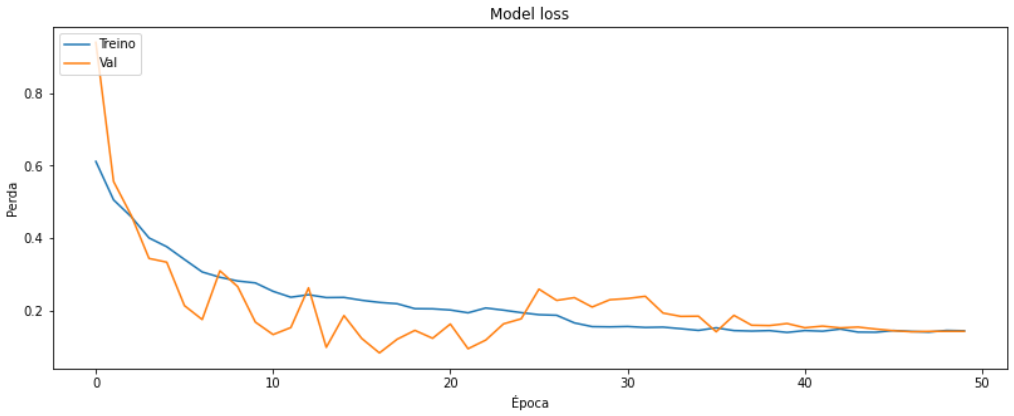
As Figuras 36 e 37 apresentam algumas das máscaras preditas pelos modelos das três arquiteturas propostas, os quais foram treinados com o *dataset* $ISIC_A$ com aumento de dados, novamente, pelo melhor desempenho dentre os demais. A Figura 36 apresenta as predições feitas a partir de imagens presentes no *dataset* ISIC, enquanto a Figura 37 as predições de imagens do *sataset* PH^2. Essas imagens mostram-se condizentes com as métricas apresentadas nas tabelas acima, sendo visualmente possível perceber que o arquitetura U-Net+VGG-19 foi a rede que obteve as piores métricas das apresentadas, embora seja também a mais sensível a amostras positivas, como também é observado nas tabelas, fazendo com que o modelo gere predições de manchas que não existem realmente.



(a) U-Net+DenseNet-121.



(b) U-Net+ResNet-50.



(c) U-Net+VGG-19.

Figura 35 – Gráficos das perdas durante o treinamento para os *datasets* de treino e validação. Eixo vertical corresponde ao valor da perda e o eixo horizontal a época.

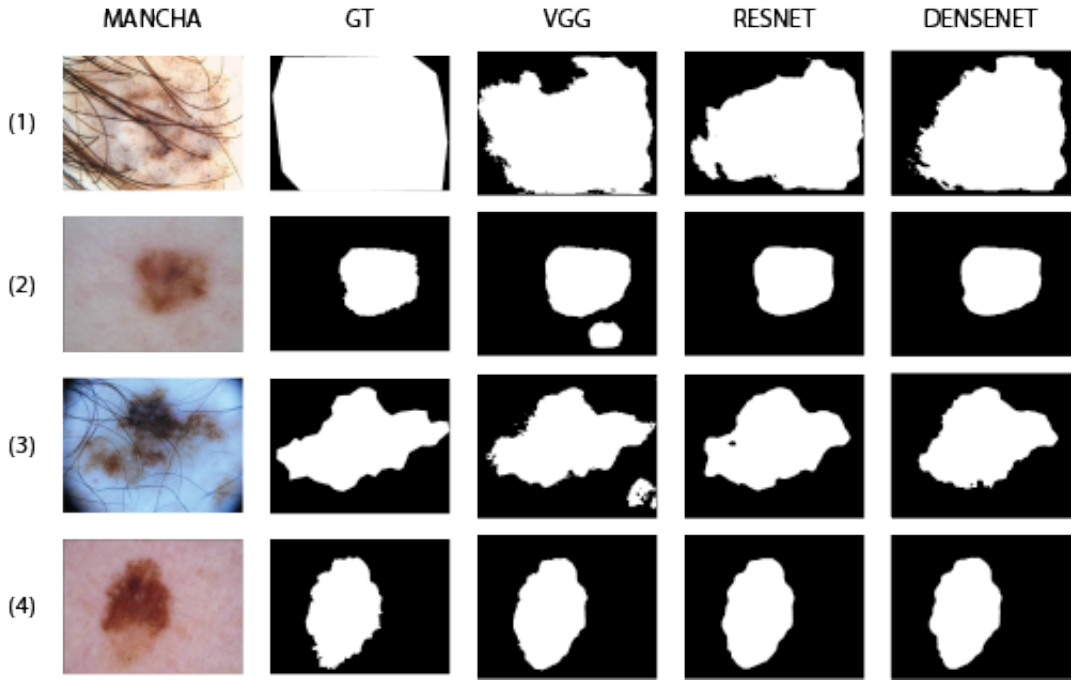


Figura 36 – Máscaras de segmentação preditas pelos modelos treinados com o *dataset* $ISIC_A$, utilizando imagens do *dataset* ISIC como teste, para as respectivas arquiteturas, manchas e GT (do inglês, *Ground Truth*) ou Verdade Fundamental.

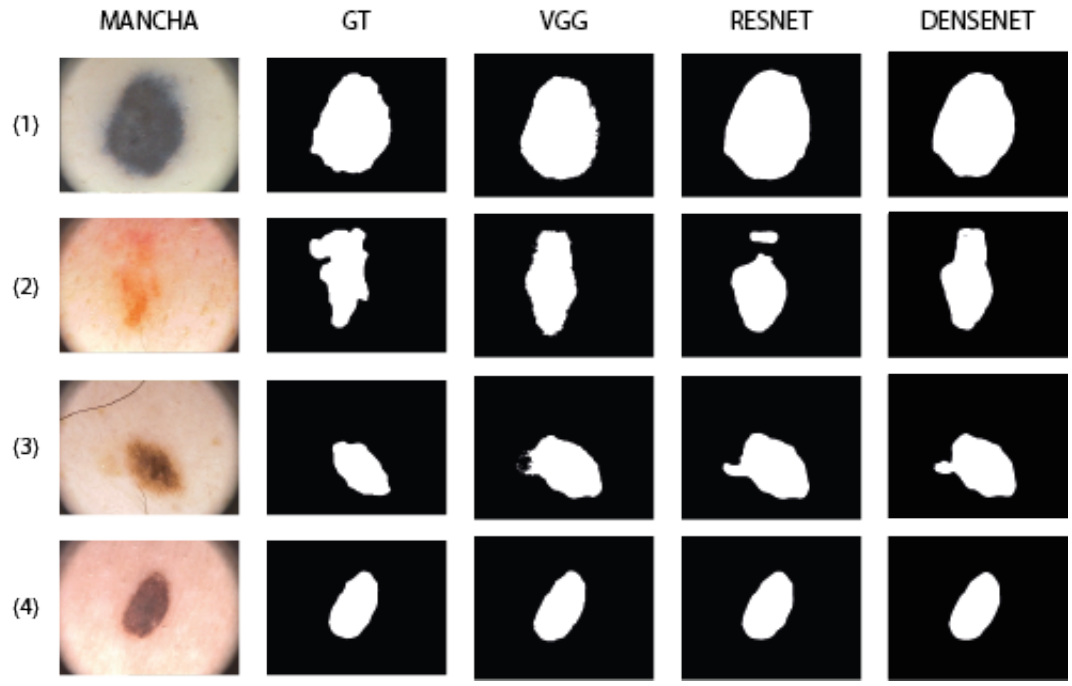


Figura 37 – Máscaras de segmentação preditas pelos modelos treinados com o *dataset* $ISIC_A$, utilizando imagens do *dataset* PH^2 como teste, para as respectivas arquiteturas, manchas e GT

Para comparação das métricas obtidas pelas redes treinadas, utilizamos as redes

citadas no Capítulo 2. Os valores para cada rede selecionada são apresentados na Tabela 12 juntamente com os resultados apresentados neste trabalho.

Tabela 12 – Comparação de resultados de arquiteturas encontradas na literatura

	Jaccard(%)	*Dice(%)*	*AC(%)*	*P(%)*	*EP(%)*	*SE(%)*
Sheng Chen et al.	78,7	86,8	94,4	-	-	-
ResNetV2	80,2	87,6	94,4	-	96,4	87,3
Ensemble	80,6	87,9	94,6	-	96,9	87,9
U-Net+VGG-19	81,00	89,48	95,47	92,73	97,96	86,46
U-Net+ResNet-50	81,94	90,07	96,42	95,33	99,01	85,37
U-Net+DenseNet-121	81,19	92,93	96,29	95,61	99,09	84,31

O trabalho de Jahanifar et al. (JAHANIFAR et al., 2018) apresentou os índices que mais se aproximaram dos resultados obtidos pelas três arquiteturas utilizadas nesse trabalho, obtendo um *Jaccard* de 80,2%, mas ainda assim não superou os índices que obtivemos na rede com menor desempenho. Porém, a sensibilidade de ambas as redes, ResNetV2 e Ensemble, apresentou resultados melhores. Essa métrica também é importante para o modelo, já que informa a porcentagem de *pixels* categorizados como melanoma que realmente pertencem ao conjunto da doença.

Além do trabalho citado acima (que apresentou melhores resultados encontrados), as redes foram comparadas ao trabalho de (CHEN et al., 2018). Embora seu processo seja mais complexo, seus índices resultantes também foram superados pelo presente trabalho.

Não foram encontradas, em nossas pesquisas, redes com resultados superiores em métricas de *Jaccard* e *Dicce* para o banco de dados testado.

5 Conclusões e Projetos Futuros

Este trabalho teve como objetivo um estudo comparativo entre arquiteturas de AP para a segmentação de imagens dermatoscópicas. A DenseNet-121, ResNet-50 e Vgg-19 foram utilizadas como *backbones* na etapa de *encoder* da U-Net. Cada arquitetura foi treinada e testada utilizando 3 *datasets* diferentes, gerando assim um modelo a cada teste. Por fim as métricas e máscaras de segmentação preditas pela rede foram obtidas. Com a realização destes experimentos, surgiram problemas e algumas soluções possíveis.

O principal problema encontrado, durante a realização dos teste, foi a escolha dos hiperparâmetros, tendo em vista que não se tem um número exato de épocas, tamanho do *batch* e taxa de aprendizado. Sendo assim foram necessários várias tentativas de ajuste, sempre observando o comportamento das métricas no decorrer do treinamento. A arquitetura U-Net+VGG-19, por exemplo, apresentou métricas semelhantes as demais, utilizando apenas metade das épocas, pois foi constatado no treinamento que o aprendizado, após o período de 50 épocas, deixava de ser significativo. Já a U-Net+DenseNet-121 apresentou melhores resultados utilizando um *batch* maior que a demais. Portanto a escolha dos hiperparâmetros deu-se de forma empírica.

Apesar da U-Net+ResNet-50 ter apresentado melhor valor médio na métrica *Jaccard* considerando todos os *datasets* testados, a U-Net+DenseNet-121 mostrou melhor desempenho se considerada o valor absoluto dos testes, a média sem aumento de dados, o treino com o *dataset* PH^2 e os gráficos gerados.

A partir da realização deste trabalho foi possível constatar a notoriedade do uso da arquitetura U-Net na tarefa de segmentação de imagens. A utilização dos *backbones* atingiu resultados adequados na tarefa de segmentação, levando em consideração que a segmentação é uma etapa extremamente importante e, consequentemente, difícil de ser feita.

5.1 Trabalhos futuros

Como sugestão de trabalhos futuros, foram levantados os seguintes itens.

- Testar as redes apresentadas nesse trabalho para bancos de dados mais recentes;
- Utilizar outras técnicas para obter uma binarização mais adequada das máscaras preditas pela rede;
- Aplicação de outras técnicas de validação cruzada;

- Utilizar outras técnicas de segmentação, como a segmentação em conjunto (ou, do inglês, *emsemble*), que utiliza as máscaras obtidas por diferentes arquiteturas e mesclam ambas as máscaras para a obtenção de novos resultados, ou até combinar a segmentação da mancha de pele com a segmentação da pele (invertendo-se as máscaras). Esse tipo de segmentação geralmente apresenta resultados superiores.

Referências

ACS. *Key Statistics for Melanoma Skin Cancer*. 2020. Disponível em: <https://www.cancer.org/cancer/melanoma-skin-cancer/about/key-statistics.html>. Citado na página 21.

ACS. *Survival Rates for Melanoma Skin Cancer*. 2020. Disponível em: <https://www.cancer.org/cancer/melanoma-skin-cancer/detection-diagnosis-staging/survival-rates-for-melanoma-skin-cancer-by-stage.html>. Citado na página 21.

ALBAWI, S.; MOHAMED, T. A.; AL-ZAWI, S. Understanding of a convolutional neural network. In: *2017 International Conference on Engineering and Technology (ICET)*. [S.l.: s.n.], 2017. p. 1–6. Citado na página 46.

BRESLOW, A. Thickness, cross-sectional areas and depth of invasion in the prognosis of cutaneous melanoma. *Annals of surgery*, Lippincott, Williams, and Wilkins, v. 172, n. 5, p. 902, 1970. Citado na página 27.

CHEN, S. et al. A multi-task framework with feature passing module for skin lesion classification and segmentation. In: IEEE. *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. [S.l.], 2018. p. 1126–1129. Citado 3 vezes nas páginas 56, 57 e 74.

CHOLLET, F. *Deep Learning with Python*. [S.l.]: MANNING SHELTER ISLAND, 2018. ISBN 9781617294433. Citado 7 vezes nas páginas 11, 30, 31, 37, 38, 39 e 40.

CONTE, J. *Instituto vencer o câncer - Melanomas são responsáveis por 75% das mortes por câncer de pele*. 2019. Disponível em: <https://vencerocancer.org.br/noticias-melanoma/melanomas-sao-responsaveis-por-75-das-mortes-por-cancer-de-pele/#imageclose-5685>. Citado na página 21.

DONG, M. et al. Sparse fully convolutional network for face labeling. *Neurocomputing*, v. 331, p. 465 – 472, 2019. ISSN 0925-2312. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0925231218314267>. Citado na página 52.

FILHO, O. M.; NETO, H. V. *Processamento digital de imagens*. [S.l.]: Brasport, 1999. Citado na página 29.

FRIEDMAN, R. J.; RIGEL, D. S.; KOPF, A. W. Early detection of malignant melanoma: the role of physician examination and self-examination of the skin. *CA: a cancer journal for clinicians*, Wiley Online Library, v. 35, n. 3, p. 130–151, 1985. Citado na página 27.

GHAHREMANI, P.; DROPPO, J.; SELTZER, M. Linearly augmented deep neural network. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.: s.n.], 2016. p. 5085–5089. Citado na página 49.

GONZALEZ, R. C.; WOODS, R. E. *Processamento de imagens digitais*. [S.l.]: Pearson, 2009. v. 3. Citado 2 vezes nas páginas 28 e 29.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado na página 43.

GOYAL, M. et al. Skin lesion segmentation in dermoscopic images with ensemble deep learning methods. *IEEE Access*, v. 8, p. 4171–4181, 2020. Citado na página 59.

GURNEY, K. *An introduction to neural networks*. [S.l.]: Taylor Francis e-Library, 2004. ISBN 0-203-45151-1. Citado 5 vezes nas páginas 11, 33, 35, 36 e 37.

HAN, S. S. et al. Classification of the clinical images for benign and malignant cutaneous tumors using a deep learning algorithm. *Journal of Investigative Dermatology*, v. 138, n. 7, p. 1529 – 1538, 2018. ISSN 0022-202X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0022202X18301118>. Citado na página 22.

HAO, S.; ZHOU, Y.; GUO, Y. A brief survey on semantic segmentation with deep learning. *Neurocomputing, Elsevier*, v. 406, p. 302–321, 2020. Citado na página 22.

HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. [S.l.]: Prentice Hall, 1999. Citado 2 vezes nas páginas 32 e 34.

HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. Citado 4 vezes nas páginas 12, 49, 50 e 61.

H.KITTLER et al. Computational methods for the image segmentation of pigmented skin lesions: A review. *Computer Methods and Programs in Biomedicine. Elsevier*, v. 3, n. 3, p. 159–165, 2017. Citado na página 22.

HUANG, G. et al. Densely connected convolutional networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017. Citado 4 vezes nas páginas 12, 50, 51 e 62.

INCA. *Câncer de pele não melanoma*. 2020. Disponível em: <https://www.inca.gov.br/tipos-de-cancer/>. Citado 2 vezes nas páginas 21 e 25.

INCA. *Câncer de pele melanoma*. 2020. Disponível em: <https://www.inca.gov.br/tipos-de-cancer/cancer-de-pele-melanoma>. Citado na página 25.

ISIC. *ISIC Challenge*. 2020. Disponível em: <https://challenge.isic-archive.com/>. Citado na página 59.

JAFARI, M. et al. Skin lesion segmentation in clinical images using deep learning. *International Conference on Pattern Recognition (ICPR)*, v. 23rd, 2016. Citado na página 22.

JAHANIFAR, M. et al. Segmentation of skin lesions and their attributes using multi-scale convolutional neural networks and domain specific augmentations. *arXiv preprint arXiv:1809.10243*, 2018. Citado 2 vezes nas páginas 55 e 74.

JAHANIFAR, M. et al. *Segmentation of Skin Lesions and their Attributes Using Multi-Scale Convolutional Neural Networks and Domain Specific Augmentations*. 2019. Citado na página 66.

JAHNE, B. Digital image processing: concepts. *Algorithms, and Scientific*, 1997. Citado na página 29.

JOGIN, M. et al. Feature extraction using convolution neural networks (cnn) and deep learning. In: *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*. [S.l.: s.n.], 2018. p. 2319–2323. Citado na página 46.

JORDAN, J. *An overview of semantic image segmentation.* 2018. Disponível em: <https://www.jeremyjordan.me/semantic-segmentation/>. Citado 2 vezes nas páginas 11 e 30.

KHAN, S. et al. [S.l.: s.n.], 2018. Citado 12 vezes nas páginas 11, 31, 32, 39, 41, 42, 43, 44, 45, 46, 63 e 65.

KLEBANOV, N. et al. Clinical spectrum of cutaneous melanoma morphology. *Journal of the American Academy of Dermatology*, Elsevier, v. 80, n. 1, p. 178–188, 2019. Citado 2 vezes nas páginas 11 e 26.

LAMESKI, J. et al. Skin lesion segmentation with deep learning. In: *IEEE EUROCON 2019 -18th International Conference on Smart Technologies*. [S.l.: s.n.], 2019. p. 1–5. Citado na página 59.

LECUN, Y. et al. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, v. 1, n. 4, p. 541–551, 1989. Citado na página 41.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998. Citado 2 vezes nas páginas 12 e 48.

LEÓN, M. V. et al. Melanoma. *Medicine-Programa de Formación Médica Continuada Acreditado*, Elsevier, v. 11, n. 26, p. 1597–1607, 2013. Citado na página 27.

Lin, T.-Y. et al. Focal Loss for Dense Object Detection. *arXiv e-prints*, p. arXiv:1708.02002, ago. 2017. Citado 3 vezes nas páginas 12, 65 e 66.

MARSLAND, S. *MACHINE LEARNING - An Algorithmic Perspective*. 2. ed. [S.l.]: CRC Press - Chapman Hall, 2014. ISBN 978-1466583283. Citado na página 31.

MATHWORKS. *Convolutional Neural Network - 3 things you need to know.* 2020. Disponível em: <https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html>. Citado 2 vezes nas páginas 11 e 41.

MAYER, J. Systematic review of the diagnostic accuracy of dermatoscopy in detecting malignant melanoma. *Medical Journal of Australia*, v. 167(4), p. 206–210, 1997. Citado na página 22.

MENDONÇA, T. et al. Ph2 - a dermoscopic image database for research and benchmarking. In: *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. [S.l.: s.n.], 2013. p. 5437–5440. Citado na página 59.

MENZIES, S. et al. The performance of solarscan an automated dermoscopy image analysis instrument for the diagnosis of primary melanoma. *Archives of Dermatology*, v. 141(11), p. 1388–1396, 2005. Citado 2 vezes nas páginas 22 e 30.

MISSINGLINK.AI. *Fully Connected Layers in Convolutional Neural Networks: The Complete Guide*. 2020. Disponível em: <https://missinglink.ai/guides/convolutional-neural-networks/fully-connected-layers-convolutional-neural-networks-complete-guide/>. Citado 3 vezes nas páginas 12, 46 e 47.

MOREAU, J. F.; WEISSFELD, J. L.; FERRIS, L. K. Characteristics and survival of patients with invasive amelanotic melanoma in the usa. *Melanoma research*, LWW, v. 23, n. 5, p. 408–413, 2013. Citado na página 27.

MUHAMMAD, U. et al. Pre-trained vggnet architecture for remote-sensing image scene classification. In: *2018 24th International Conference on Pattern Recognition (ICPR)*. [S.l.: s.n.], 2018. p. 1622–1627. Citado 2 vezes nas páginas 12 e 49.

MULLER, B.; REINCHARDT, J.; STRICKLAND, M. T. *Neural Networks An Introduction*. [S.l.]: Springer, 1995. ISBN 978-3-540-60207-1. Citado 4 vezes nas páginas 11, 32, 33 e 35.

NAKAURA, T. et al. A primer for understanding radiology articles about machine learning and deep learning. *Diagnostic and Interventional Imaging*, 2020. Citado 2 vezes nas páginas 31 e 39.

OLIVEIRA, R. B. et al. Diagnostic accuracy of dermoscopy. *The Lancet Oncology, Elsevier*, v. 131, p. 127–141, 2016. Citado na página 22.

OMS. *How common is skin cancer?* 2020. Disponível em: <https://www.who.int/uv/resources/FAQ/skincancer/en/index1.html>. Citado na página 21.

ONCOGUIA. *Taxa de Sobrevida para Câncer de Pele Melanoma*. 2020. Disponível em: <http://www.oncoguia.org.br/conteudo/taxa-de-sobrevida-para-cancer-de-pele-melanoma/7062/187/>. Citado na página 21.

OSTROWSKI, S. M.; FISHER, D. E. Melanosome transfer: it is best to give and receive. *Hematol Oncol Clin N Am, Elsevier*, 2021. Citado 3 vezes nas páginas 25, 26 e 27.

OSTROWSKI, S. M.; FISHER, D. E. Melanosome transfer: it is best to give and receive. *Hematol Oncol Clin N Am, Elsevier*, 2021. Citado na página 26.

PROTA, G. Recent advances in the chemistry of melanogenesis in mammals. *The Journal of Investigative Dermatology, Elsevier*, 1980. Citado na página 26.

QUEIROZ, J. E. R. de; GOMES, H. M. Introdução ao processamento digital de imagens. *Rita*, v. 13, n. 2, p. 11–42, 2006. Citado 2 vezes nas páginas 28 e 29.

ROCHA, J. C. Cor luz, cor pigmento e os sistemas rgb e cmy. *Revista Belas Artes, São Paulo*, n. 3, p. 1–19, 2010. Citado na página 28.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: NAVAB, N. et al. (Ed.). *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Cham: Springer International Publishing, 2015. p. 234–241. Citado 3 vezes nas páginas 12, 52 e 53.

SAHA, S. *Toward data science - A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. 2018. Disponível em: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. Citado na página 43.

SANTOS, M. T. *O que é melanoma? Brasileiros desconhecem o câncer de pele mais letal*. 2019. Disponível em: <https://saude.abril.com.br/medicina/o-que-e-melanoma-brasileiros-desconhecem-o-cancer-de-pele-mais-letal/>. Citado na página 25.

SBD. *Câncer da pele*. 2020. Disponível em: <https://www.sbd.org.br/dermatologia/pele/doencas-e-problemas/cancer-da-pele/64/>. Citado na página 25.

SHAIKH, W. R.; XIONG, M.; WEINSTOCK, M. A. The contribution of nodular subtype to melanoma mortality in the united states, 1978 to 2007. *Archives of dermatology*, American Medical Association, v. 148, n. 1, p. 30–36, 2012. Citado na página 27.

SHELHAMER, E.; LONG, J.; DARRELL, T. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 39, n. 4, p. 640–651, 2017. Citado na página 52.

SILVA, L. A.; PERES, S. M.; BOSCARIOLI, C. *Introdução à mineração de daods - com aplicações em R*. [S.l.]: GEN Ltc., 2016. ISBN 853528446X. Citado na página 54.

SILVEIRA, M. et al. Comparison of segmentation methods for melanoma diagnosis in dermoscopy images. *IEEE Journal of Selected Topics in Signal Processing*, v. 3, n. 1, p. 35–45, 2009. Citado na página 22.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 09 2014. Citado 2 vezes nas páginas 49 e 61.

SMITH, L.; MACNEIL, S. State of the art in non-invasive imaging of cutaneous melanoma. *Skin Research Technology*, 2011. Citado na página 22.

SRIVASTAVA, N. et al. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, v. 15, p. 1929–1958, 06 2014. Citado 3 vezes nas páginas 12, 46 e 47.

STURM, R. A. Molecular genetics of human pigmentation diversity. *Human molecular genetics*, Oxford University Press, v. 18, n. R1, p. R9–R17, 2009. Citado na página 26.

TIAN, Y.; FU, S. A descriptive framework for the field of deep learning applications in medical images. *Knowledge-Based Systems*, v. 210, p. 106445, 2020. ISSN 0950-7051. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0950705120305748>. Citado na página 46.

VERGARA, R. F. *Detecção de alterações cerebrais anatômicas associadas à esquizofrenia com base em redes convolucionais aplicadas a imagens de ressonância magnética*. 89 p. Monografia (Pós-Graduação) — Faculdade Gama - FGA, Universidade de Brasília, Brasília, DF, 2018. Citado 2 vezes nas páginas 39 e 47.

WANG, Z. et al. Cnn explainer: Learning convolutional neural networks with interactive visualization. *Cornell University*, 2020. Citado 5 vezes nas páginas 41, 42, 43, 44 e 46.

WEINSTOCK, M.; SOBER, A. The risk of progression of lentigo maligna to lentigo maligna melanoma. *British Journal of Dermatology*, Wiley Online Library, v. 116, n. 3, p. 303–310, 1987. Citado na página 27.

WU, X.; HAMMER, J. A. Melanosome transfer: it is best to give and receive. *Current Opinion in Cell Biology, Elsevier*, 2014. Citado na página 26.

YAKUBOVSKIY, P. *Segmentation Models*. [S.l.]: GitHub, 2019. <https://github.com/qubvel/segmentation_models>. Citado na página 64.

YIN, X.; YAN, L.; SHIN, B.-S. Tgv upsampling: A making-up operation for semantic segmentation. *Computational Intelligence and Neuroscience*, Hindawi, v. 80, n. 2019, 2019. Citado 2 vezes nas páginas 11 e 45.

ZHANG, L.; HE, M.; SHAO, S. Machine learning for halide perovskite materials. *Nano Energy*, v. 78, 2020. Citado na página 32.

ZHOU, H. et al. Skin lesion segmentation using an improved snake model. In: *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. [S.l.: s.n.], 2010. p. 1974–1977. Citado na página 22.