UnB

Universidade de Brasília

Departamento de Engenharia Elétrica

SEGMENTAÇÃO SEMÂNTICA DE IMAGENS DE RESSONÂNCIA MAGNÉTICA PARA ANÁLISE DE TUMORES CEREBRAIS USANDO REDES NEURAIS ARTIFICIAIS

Autor: Vitor Martin Bordini

Orientadora: Mylène Christine Queiroz Farias

Brasil

8 de novembro de 2021

VITOR MARTIN BORDINI

SEGMENTAÇÃO SEMÂNTICA DE IMAGENS DE RESSONÂNCIA MAGNÉTICA PARA ANÁLISE DE TUMORES CEREBRAIS USANDO REDES NEURAIS ARTIFICIAIS

Trabalho de Conclusão de Curso submetido à Universidade de Brasília, como requisito necessário para obtenção do grau de Bacharel em Engenharia Elétrica

Brasília, 8 de novembro de 2021

Vítor Martin Bordini

SEGMENTAÇÃO SEMÂNTICA DE IMAGENS DE RESSONÂNCIA MAGNÉTICA PARA ANÁLISE DE TUMORES CEREBRAIS USANDO REDES NEURAIS ARTIFICIAIS

Monografia submetida ao curso de graduação em Engenharia Elétricada Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Elétrica.

Trabalho aprovado. Brasília, DF, 8 de novembro de 2021:

Professora Doutora Mylène Christine de Queiroz Farias
Orientador

Professor Doutor Daniel Guerreiro e Silva
Convidado 1

Professor Doutor Cristiano Jacques Miosso
Convidado 2

Brasília, DF
2021

Este trabalho é dedicado à minha mãe, Valéria Regina Zanardo Martin, e a todas as crianças que sonham em ser cientistas.

Agradecimentos

Agradeço:

Aos meus colegas e amigos, especialmente Gustavo Henrique de Souza Leão ,Thiago de Oliveira Magalhães, Lúcio Bragança Zago e Guilherme Raposo Diniz Vieira, , por todo o apoio.

Aos meus pais, André de Souza Bordini e Valéria Regina Zanardo Martin, por todo suporte financeiro e emocional.

À minha orientadora, Mylene Christine Queiroz de Farias, por todo o aprendizado e pela oportunidade desse projeto.

À banca, Daniel Guerreiro e Silva e Cristiano Jacques Miosso , pela disposição.

À pessoa que mais me apoiou durante esse tempo, Jackson Santos Ferreira.

"Eu não creio que exista algo mais emocionante para o coração humano do que a emoção sentida pelo inventor quando ele vê alguma criação da mente se tornando algo de sucesso. Essas emoções fazem o homem esquecer comida, sono, amigos, amor, tudo." Nikola Tesla

Resumo

Para haver uma maior chance de sobrevivência, é recomendável o diagnóstico precoce dos tumores cerebrais. Por exemplo, os glioblastomas, um dos tumores cerebrais mais agressivos, possuem uma taxa de sobrevivência menor que 20% para um período de 5 anos após o diagnóstico. Por isso, é necessário um acompanhamento médico periódico. Uma opção comum e não-invasiva para o diagnóstico é por meio de Imagens de Ressonância Magnética (*Magnetic Resonance Images* - MRIs) para a detecção de tumores.

A análise das imagens requer um especialista. No entanto, até mesmo especialistas estão sujeito a erros. Há vários relatos na literatura médica sobre erros de especialistas. Além disso, a tarefa de detectar o tumor para o diagnóstico também consome tempo. Uma forma de evitar esses erros, reduzir o tempo necessário para o diagnóstico e garantir uma maior segurança e precisão no diagnóstico é por meio da detecção automática de tumores usando técnicas de processamento de imagens e redes neurais.

Esse trabalho propõe o uso de redes neurais em MRIs de cérebros para detecção e segmentação de tumores cerebrais. Implementamos algumas das arquiteturas mais populares no banco de dados público BRATS. Esse trabalho também demonstra como a degradação tem um efeito preocupante na segmentação semântica por reduzir a qualidade das MRIs e o desempenho das redes.

Por exemplo, uma fonte comum de degradação é a movimentação do paciente durante a aquisição da imagem, que pode tornar a imagem mais degradada quanto mais movimentos o paciente faz, dependendo da rotação e translação de cada movimento. Nas simulações dessa degradação, quando aumentamos o número de movimentos de 1 para 10 do paciente, para a arquitetura menos robusta às degradações, a Residual U-net, há uma queda de aproximadamente 16% na média do coeficiente Dice de todas as classes e um aumento de 11,1 da média da distância de Hausdorff de todas as classes. Na mesma simulação, para a arquitetura mais robusta às degradações, a V-net, a queda da média do coeficiente Dice foi de 9% e o aumento da média da distância de Hausdorff foi de 3,8.

Outras degradações, como o ruído gaussiano, tiveram uma variação menor mas igualmente notável. Quando o desvio padrão do ruído gaussiano aumentou de $0,065$ para $0,65$, houve uma uma queda de 3,7% no coeficiente Dice e um aumento da distância de Hausdorff de 1,75. Para a V-net, a diminuição da média do coeficiente Dice foi de 2% e o aumento da média da distância de Hausdorff. foi de 1,2. Assim, para diminuir o desempenho da rede, é necessária uma alta intensidade da degradação.

Palavras-chave: MRI, segmentação semântica, tumor cerebral, deep learning, degradação

Abstract

Brain tumors are deadly if not treated in the right time. That is why an early diagnosis is essential. A common way to diagnose it is by performing exams with Magnetic Resonance Images (MRIs) which are later analysed by a specialist. Unfortunately, sometimes errors are made. In fact, there are numerous reports in the medical literature about those mistakes Also, the task of analysing a MRI and detecting a tumor is time-consuming. In order to avoid those mistakes and reduce the necessary time for diagnosis, we can make an automatic tumor detection system by using image processing and neural networks. Moreover, since the quality of MRIs can be affected by the acquisition, coding, storage and transmission of the images, we also need to verify how those automatic diagnose techniques are affected by noisy images, which are common in the real world.

This works presents an automatic diagnostic for brain tumors, which is implemented by some of the most popular convolution neural networks architecture. The system was trained and tested on the BRATS dataset. We also demonstrate the effect of degradation in their performance.

For example, if a patient moves during the MRI acquisition, the MRI is degraded. The intensity of this degradation could increase if we increase the number of movements the patient has done, depending on the translation and rotation of each movement. In our simulations, when we increase the number of movements from 1 to 10, for the Residual U-net, the least robust architecture studied, there is a drop of 16% on the average Dice score and an increase of 11.1 on the average Hausdorff distance. On the same simulation, for V-net, the most robust architecture, there is a drop of 9% on the average Dice score and an increase of 3.8 on the average Hausdorff distance.

Also, the Gaussian noise has a smaller drop but still remarkable. When we increase the Gaussian Noise standard deviation from 0.065 to 0.65, for the Residual U-net, there is a drop of 3.7% on the average Dice score and an increase of 1.7 on the average Hausdorff distance. On the same simulation, for V-net, the most robust architecture, there is a drop of 2% on the average Dice score and an increase of 1.2 on the average Hausdorff distance. Thus, from the experiments, it is easy to realize that the networks are not sensible to those problems. However, motion during image extraction is the source that causes the most problems.

Keywords: MRI, semantic segmentation, brain tumor, deep learning, degradation

Sumário

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

MRI: Magnetic Resonance Imageing

ANN: Artificial Neural Network

CNN: Convolution Neural Network

SVM: Support Vector Machine

KNN: K-nearest neighbors

SGD: Stochastic Gradient Descent

DMFnet: Dilated Multi-Fiber Net

PSNR: Peak-to-Signal Noise Ratio

Rmsprop: Root Mean Square Propagation

1 Introdução

1.1 Contexto

O cérebro é um dos órgãos mais importantes do corpo humano. Esse órgão é responsável pelo controle de mecanismos voluntários e involuntários de todos os demais sistemas do organismo [Raichle 2010]. Isso justifica a necessidade de manter a saúde desse órgão e se precaver contra possíveis doenças. Nesse sentido, uma grande ameaça à sua saúde são os tumores cerebrais, que podem ser de dois tipos:

- Benignos: Eles consistem em uma massa de células que cresce indefinidamente. Contudo, são menos agressivos e não são capazes de se espalhar pelo organismo. Esses tumores possuem as seguintes subclassificações dependendo da sua origem [Roy et al. 2013]:
  - Meningioma: Originários na meninge.
  - Craniofaringeoma e Adenoma pituitário: Origem na hipófise, principal glândula do cérebro.
- Malignos: Também conhecidos como cancerígenos, esses tumores crescem rápido e são mais agressivos que os tumores benignos. Além disso, esses tumores são capazes de invadir tecidos próximos, o que os torna mais preocupantes. Por isso, eles podem se apresentar diretamente no cérebro (primário) ou serem originários de outro órgão do corpo e se espalhar para lá (metastático). Quanto à origem no cérebro, há três tipos de tumores malignos [Roy et al. 2013]:
  - Meduloblastoma: Comum em crianças, são tumores na fossa central.
  - Glioma: Originário das células gliais, é um tipo comum de câncer. Podem ser classificados em astrocitoma, oligoastrocitoma e glioblastomas.
  - Linfoma: É o câncer originário no sistema linfático.

A Figura 1 apresenta um gráfico que relata a taxa de sobrevivência em período de até 15 anos após o diagnóstico. Os tumores menos agressivos possuem taxa de sobrevivência maior que 70%, enquanto que os tumores mais agressivos possuem taxas bem menores. O Glioblastoma, por exemplo, possui uma taxa de sobrevivência menor que 20 % em menos de 5 anos depois do diagnóstico.

Uma característica importante desses tumores é sua raridade. Eles representam apenas o décimo-sétimo câncer mais frequente. Isso dificulta o seu estudo pois faltam

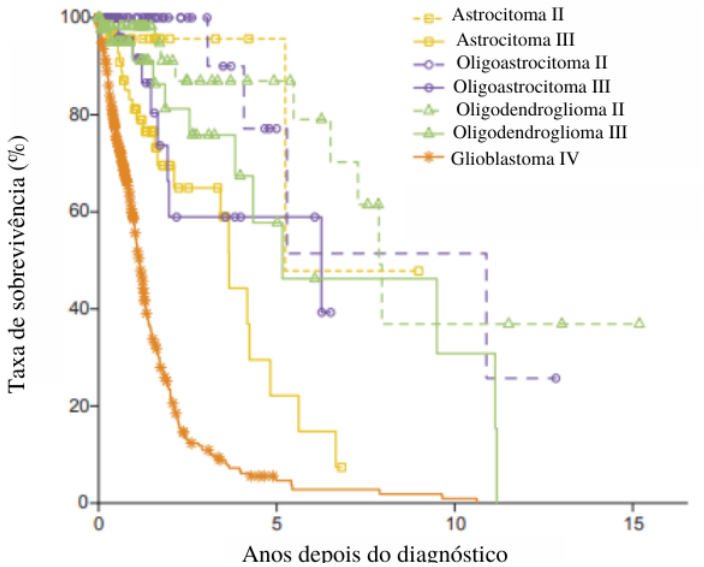


Figura 1 – Sobrevivência depois do diagnóstico para tipos diferentes de gliomas [WILD, WEIDERPASS e STEWART 2020]. Percebe-se que o glioblastoma e o astrocitoma (linha laranja e amarela, respectivamente) possuem uma taxa menor que de 20% depois de 5 anos, enquanto tumores menos agressivos, como o oligoastrocitoma e o oligodendroglioma (linha roxa e verde, respectivamente) possuem taxas acimas de 60 % no mesmo período.

dados para uma análise robusta. Mesmo assim, há um consenso que o diagnóstico precoce é essencial para que haja uma boa probabilidade de sobrevivência do indivíduo [Zhao e Jia 2015].

Uma opção comum e não-invasiva para o diagnóstico é por meio de MRIs (Imagens de Ressonância Magnética, em Inglês). No MRI, os campos eletromagnéticos excitam prótons em átomos de hidrogênio que compõem moléculas de água, abundante nos tecidos humanos. Isso cria um sinal detectável, que é codificado na forma de imagens. O MRI está entre os exames mais populares no diagnóstico de câncer cerebral devido ao alto contraste da imagem em relação a outros exames de imageamento com uma resolução comparável a outras alternativas em imageamento, como por exemplo a tomografia computadorizada (TC). Consequentemente, esse exame garante uma boa precisão do diagnóstico e uma segurança ao paciente [Yang et al. 2018]. A resolução e o contraste de MRIs, no entanto, podem ser prejudicados se degradações forem introduzidas no processo de aquisição, codificação, transporte e armazenamento da imagem. Em imagens com degradações a visualização de um tumor pode ser mais difícil.

Dados todos esses aspectos, é essencial que as pessoas tenham um acompanhamento médico periódico de forma a permitir um diagnóstico precoce e um tratamento mais eficaz desse tipo de câncer. No entanto, até mesmo especialistas em diagnóstico de tumor cerebral podem errar. Há vários relatos de erros médicos na literatura [Jorritsma, Cnossen e van Ooijen 2015]. Além disso, a detecção do tumor é uma tarefa que consome tempo [Dong

et al. 2017]. Por essas duas razões (possível erro no diagnóstico e consumo de tempo), o auxílio de uma máquina com um desempenho aceitável na tarefa de detectar tumores é desejável tanto do ponto de vista dos pacientes, que obtêm um diagnóstico mais preciso, quanto dos médicos, que têm mais segurança na hora de recomendar o melhor tratamento para os pacientes.

Recentemente, a popularidade de técnicas para detecção automática de tumores usando segmentação semântica aumentou consideravelmente. Entre os vários métodos de classificação, podemos citar: *Support Vector Machine* (SVM), *Random Forest*, *K-nearest neighbors* (KNN) e *Artificial Neural Networks* (ANN), ou redes neurais. De acordo com Vaishali Tyagi [Tyagi 2019], as redes neurais são os melhores classificadores em duas condições: presença de mais de duas classes e bancos de dados suficientemente grandes. Ambos os critérios são vistos nesse trabalho (para mais detalhes veja a seção 3.2). Por esta razão decidimos usar esse método nos experimentos deste trabalho.

Contudo, a literatura científica muitas vezes foca apenas no desempenho dos classificadores nos bancos de dados. Dessa forma, a metodologia desses trabalhos negligencia a possibilidade de haver uma redução da qualidade da imagem (uma degradação) durante os processos de extração, codificação, armazenamento e envio das imagens. O diferencial desse projeto é que além de estudar as redes neurais que detectam tumores, também se estuda como a redução da qualidade da imagem afeta o desempenho de modelo.

1.2 Definição do problema

Nesse trabalho, o problema pode ser resumido ao uso de segmentação semântica para determinar a localização do tumor no cérebro. A segmentação semântica é definida como a classificação pixel a pixel [Li, Johnson e Yeung 2017]. As classes podem ser definidas como todas as categorias de objetos que existem em um banco de dados. A Figura 2 mostra um exemplo dessa tarefa onde o algoritmo tenta classificar o tipo de objeto na cena. Nesse caso, pode-se notar as seguintes classes: árvore, grama, céu, gato e vaca.

1.3 Objetivos do projeto

Nesse contexto, os objetivos desse trabalho são os seguintes:

- Pesquisar as arquiteturas mais recentes para a segmentação semântica de tumores cerebrais, comparando os seus desempenhos.
- Identificar as fontes mais comuns de degradações e simulá-las.
- Verificar a sensibilidade de cada arquitetura para cada tipo de degradação de forma a determinar qual degradação prejudica a segmentação semântica.

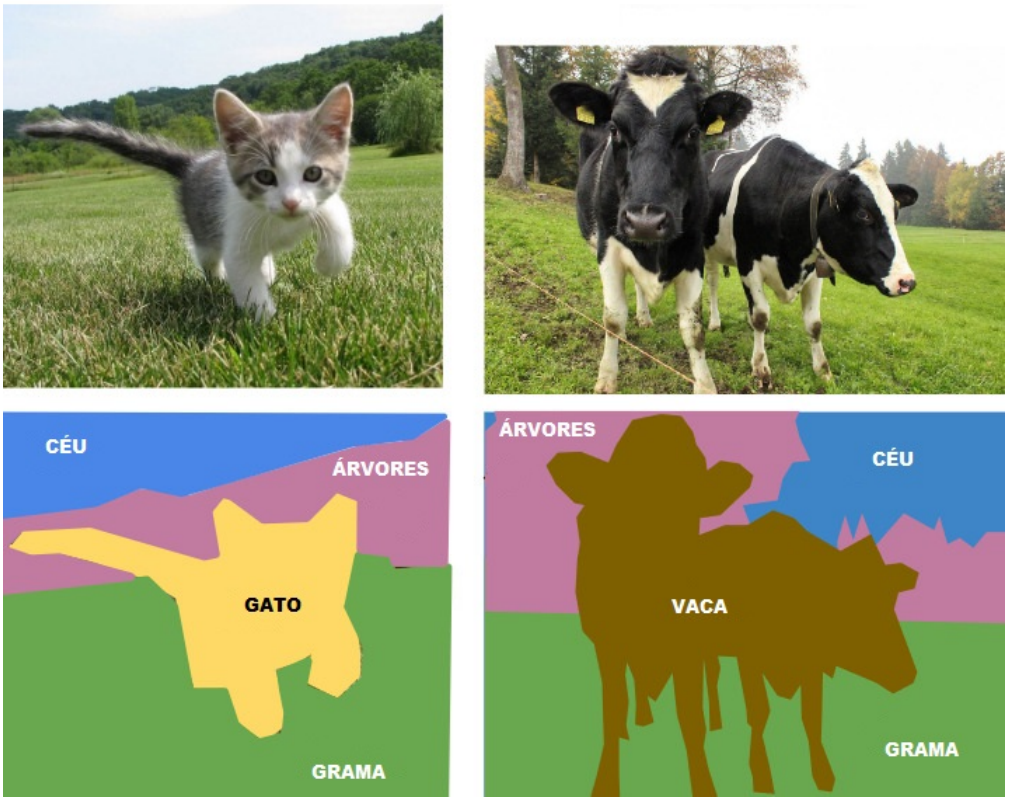


Figura 2 – Dois exemplos de segmentação semântica [Li, Johnson e Yeung 2017]. Na primeira imagem do canto superior esquerdo vê-se um gato num campo. As classes possíveis são: gato, árvore, grama e céu. Para a segunda imagem (canto superior direito) ,observam-se vacas num campo. Nesse caso, as classes possíveis são: vaca , árvore, grama e céu. A parte inferior mostra as segmentações ideais de cada uma das imagens.

2 Breve introdução a redes neurais artificiais

Antes de explicar as redes neurais, é importante entender conceitos básicos de aprendizado de máquinas pois uma rede neural é uma forma de aprendizado de máquinas. De forma geral, pode-se dividir o aprendizado de máquina em dois tipos [Goodfellow, Bengio e Courville 2016]:

- Aprendizagem não-supervisionada: É a aprendizagem onde não há uma resposta esperada. Assim, não é possível determinar se a resposta oferecida pelo modelo é adequada ou não. Um exemplo desse tipo de aprendizagem é a tarefa de dividir os dados em diversos grupos, conhecida como *clustering*.
- Aprendizagem supervisionada: É a aprendizagem onde a resposta esperada (conhecida como *label*) existe. A segmentação semântica é um exemplo desse tipo de aprendizagem, pois para saber se a previsão do modelo é adequada, deve-se compará-la ao resultado esperado fornecido pelo banco de dados. Nesse trabalho, por se tratar de uma segmentação semântica, toda a aprendizagem é supervisionada.

Além disso, as redes neurais requerem a otimização de parâmetros e hiper-parâmetros para apresentarem um bom desempenho. Logo, é importante conhecer esses conceitos para entender melhor a aprendizagem das rede neurais:

- Parâmetros: São variáveis que mudam de valor conforme o algoritmo de aprendizagem avança. O objetivo de qualquer modelo de aprendizagem supervisionada é obter os parâmetros otimizados, ou seja, aqueles que maximizam o desempenho do modelo para um determinado banco de dados.
- Hiper-parâmetros: São variáveis pré-definidas pelo cientista de dados que não são alteradas pelo algoritmo de aprendizagem. A escolha de um conjunto de hiper-parâmetros adequado ao banco de dados usado é essencial para garantir um bom desempenho pois esses hiper-parâmetros influenciam diretamente na forma como o modelo aprende as características de um banco de dados. Contudo, essa escolha não é trivial, por isso vários algoritmos foram propostos para ajudar nessa tarefa [Liashchynskyi e Liashchynskyi 2019]. O algoritmo de escolha de hiper-parâmetros mais comum é a validação cruzada, que consiste em escolher diversos valores possíveis para cada hiper-parâmetro e testar todas as combinações. Assim, a combinação que tiver o melhor desempenho é escolhida.

O aprendizado de máquina é divido em três sub-tarefas: treinar o modelo (otimizar os parâmetros), validá-lo (verificar a escolha de hiper-parâmetros) e testá-lo (verificar o

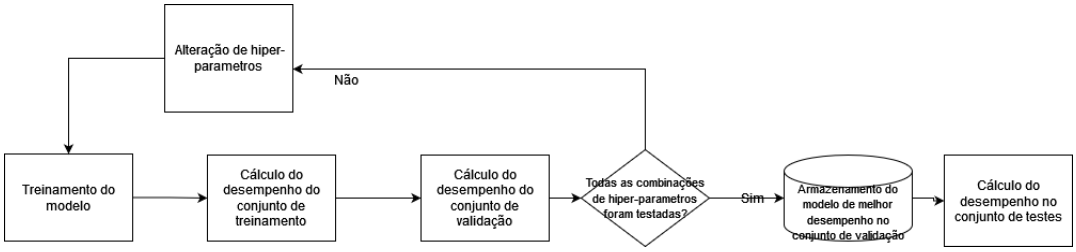


Figura 3 – Esquema simplificado da rotina de treinos das técnicas de aprendizado de máquinas. Antes de iniciar a rotina, escolhe-se um conjunto combinações de hiper-parâmetros a serem testados. A rotina é a seguinte: O modelo é treinado no conjunto de treinamento (otimização de parâmetros) para um determinado hiper-parâmetro; depois, é verificado seu desempenho no conjunto de validação. Esse processo é repetido alterando-se o hiper-parâmetro até esgotar todas as combinações do conjunto de hiper-parâmetros. Por fim, o modelo é com maior desempenho no conjunto de validação é armazenado e testado posteriormente no conjunto de testes.

desempenho do modelo). Cada uma dessas sub-tarefas é realizada em um subconjunto de dados próprio. Assim, para otimizar os parâmetros, usa-se o conjunto de treino; para verificar se a escolha de hiper-parâmetros foi adequada, o conjunto de validação; e para testar o modelo, o conjunto de teste. A rotina para a criação de um modelo é apresentada na Figura 3.

2.1 Modelo

As redes neurais artificiais (*Artificial Neural Networks* - ANN, em inglês) são modelos matemáticos capazes de prever uma determinada saída y_p por meio de uma entrada x. Essas técnicas se tornaram populares na área de saúde, especialmente no diagnóstico [Tyagi 2019]. Esse capítulo introduz brevemente o assunto [Goodfellow, Bengio e Courville 2016].

O modelo de redes neurais é inspirado no funcionamento do cérebro humano, por isso o nome. A sua unidade básica é um neurônio que tem como entrada um vetor de dados x e possuí uma única saída u_i [Guresen e Kayakutlu 2011]. Seu esquemático é mostrado na Figura 4. Esse modelo pode ser expresso pela equação abaixo:

$$u_i = \phi(w^T x + b) = \phi\left(\sum_i w_i x_i + b\right), \tag{2.1}$$

em que w é um vetor de pesos, $b \in R$ é o viés, e ϕ é uma função de ativação. As funções de ativação mais comuns são tangente hiperbólica (tanh), função logística e ReLU. Essas quatro funções são mostradas nas equações abaixo:

$$\phi(x) = \frac{e^x + e^{-x}}{e^x - e^{-x}}, \tag{2.2}$$

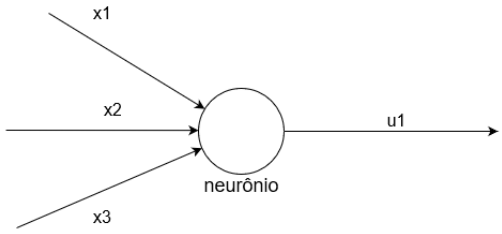


Figura 4 – Esquemático de um neurônio. Nesse caso, a entrada é um vetor de três elementos, x_1,x_2 e x_3, e a saída u_1 possui apenas um elemento. A saída e a entrada estão relacionadas pela equação 2.1

$$\phi(x) = \frac{1}{1+e^{-x}}, \tag{2.3}$$

$$\phi(x) = \max\{0, x\}. \tag{2.4}$$

Os neurônios de uma ANN são organizados em camadas. Uma camada de tamanho n gera portanto um vetor $u = [u_1, u_2, ..., u_n]$ que serve de entrada para a próxima camada e assim em diante, de camada a camada, até terminar toda a rede. Um esquemático do processo é mostrado na Figura 5.

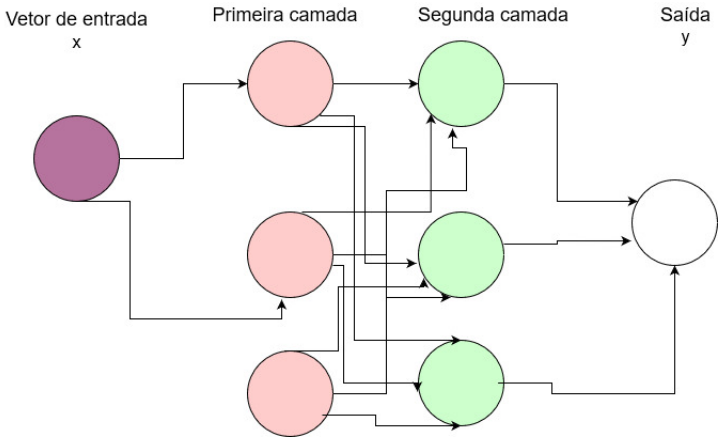


Figura 5 – Exemplo de rede neural. Essa rede é composta de duas camadas, que transformam o vetor de entrada x no vetor de saída y.

2.1.1 Funções de custo

Como já mencionado, uma resposta y_p é gerada no final da rede. Essa resposta é comparada com a resposta esperada y_l por meio da função de custo J (*loss function*, em inglês). Essa função pode variar de acordo com a tarefa usada. Seguem alguns exemplos das funções de custo mais comuns:

- Entropia cruzada: Muito utilizada em problemas de classificação, A entropia cruzada é definida pela seguinte equação [Panchapagesan et al. 2016]:

$$J = -\sum_{i=1}^{m} y_{l,i} log(p_i)), \tag{2.5}$$

em que p_i representa a probabilidade da i-ésima observação ter a classe $y_{l,i}$ desejada. Essa probabilidade é calculada via a função softmax, onde para cada i-ésima observação, o modelo gera C valores como saídas (um para cada classe). Assim, a função softmax é calculada por meio da seguinte equação:

$$p_i = \frac{e^{y_{p,i}}}{\sum_{j=1}^{C} e^{y_{p,j}}}. \tag{2.6}$$

- Função de custo L1 suave: Uma família de funções usada para a classificação [Wu et al. 2020], cuja equação é dada por:

$$J(y_{pi}, y_{li}) = \begin{cases} \frac{(y_{pi}-y_{li})^2}{2\delta}, & \text{se } x \leq \delta, \\ |y_{pi} - y_{li}| - \frac{\delta}{2}, & \text{se } x > \delta. \end{cases}$$

Nesta equação, δ é um hiper-parâmetro e i representa a i-ésima classe.

- Função de custo Dice: O Dice é uma métrica definida como a razão entre a intersecção de dois conjuntos $\hat{V}$ e V (respectivamente, os valores preditos e os verdadeiros de uma imagem) e a soma da cardinalidade de ambos. O coeficiente de Dice é definido através da seguinte equação [Fidon et al. 2018]:

$$D = \frac{2|\hat{V} \cap V|}{|\hat{V}| + |V|}. \tag{2.7}$$

Para ser usada como função de custo, no entanto, precisa-se fazer uma adaptação de forma a tornar a função diferenciável. De acordo com Fidon *et al.* [Fidon, Ourselin e Vercauteren 2021], a função mais genérica para o problema de segmentação semântica de tumores cerebrais é:

$$J = \frac{2\sum_{v \neq i}\sum_{j} p_{j,v}(1 - W^M(p_j, \hat{p}_j))}{2\sum_{v \neq i}\sum_{j} p_{j,v}(1 - W^M(p_j, \hat{p}_j)) + \sum_i W^M(p_j, \hat{p}_j))}, \tag{2.8}$$

em que p_j e $\hat{p}_j$ são as probabilidades softmax do valor verdadeiro e da predição no j-ésimo dado, respectivamente, para uma i-ésima classe. W^M é a distância Wasserstein, definida em Fidon *et al.* [Fidon et al. 2018] como um problema de otimização, no qual se minimiza o custo de transformar um vetor de probabilidade p em outro vetor $\hat{p}$. Como esses vetores de probabilidade estão ligados diretamente com as classes do banco de dados, é importante notar que para todo $i, j \in L$ (o conjunto de todas as classes) o custo de mover i para j é definida pela matriz de na matriz de distância M dos dois conjuntos $\hat{V}$ e V. A matriz de distância para dois conjuntos $\hat{V}$ e V é

definida como a matriz cujo elemento m_{ij} é igual a distância euclidiana desses dois elementos, ou seja:

$$m_{i,j} = ||v_i - \hat{v}_j||_2, \tag{2.9}$$

em que $v_i \in V$ e $\hat{v}_j \in \hat{V}$.

2.1.2 Épocas, Tamanho do Lote e Parada mais cedo

Alguns conceitos fundamentais para o correto entendimento do processo de aprendizagem de uma rede neural são as épocas, o tamanho do lote (*batch size*, em inglês) [Smith 2018] e a parada prematura do treinamento [Lin 2008], que definimos a seguir:

- Lote: Um conjunto de dados é muitas vezes grande demais para ser usado ao mesmo tempo no treinamento de uma rede. Assim, ele é subdividido em conjuntos menores chamados de lotes. No caso do problema desse trabalho, por exemplo, um lote é um subconjunto de imagens de tumor. Logo, o tamanho do lote, mais conhecido como *batch size* do inglês, se refere ao número de imagens usadas ao mesmo tempo.

- Época: Uma época é definida como a passagem de todo o conjunto de treinamento pelo modelo. Como esses dados são passados em forma de lotes escolhidos aleatoriamente, pode-se redefinir esses lotes a cada época, melhorando a performance do modelo. Portanto, a análise de uma rede é sempre feita em termos do número de épocas utilizadas.

- Parada prematura (*Early Stop*, do inglês): Essa técnica parte do principio que se depois de um determinado número de épocas a saída da função de custo do conjunto de validação não diminuir, então não haverá reduções significativas ao continuar o treinamento. Dessa forma, a continuação do treinamento poderá levar a uma super-aprendizagem do modelo. Em outras palavras, se a função de custo não estiver decrescendo mais, o treinamento pode ser interrompido.

2.2 Aprendizagem

A aprendizagem pode ser vista como um problema de otimização, visto que o seu principal objetivo é minimizar a função de custo J em relação aos pesos e ao viés dos neurônios. Na literatura, vários algoritmos foram propostos para resolver esse problema [Kingma e Ba 2017] [Choi et al. 2020] [Graves 2014]. Essa classe de algoritmos é chamada de otimizadores. É importante ressaltar que otimizadores mais complexos necessitam de mais hiper-parâmetros. Logo, encontrar os valores ótimos desses hiper-parâmetros é mais desafiador para otimizadores mais complexos.

2.2.1 Super e sub-aprendizagem

Os fenômenos de super e sub-aprendizagem (*over* e *underfitting*, em inglês) são caracterizados por um desempenho baixo do modelo nos conjuntos de teste e validação, sendo uma consequência de um processo de aprendizagem mal executado. Mais especificamente, na super-aprendizagem, o modelo possui um desempenho excelente no conjunto de treino, mas não consegue generalizar as predições ara dados de teste fora desse conjunto. Na sub-aprendizagem, no entanto, o modelo não tem performance aceitável em nenhum conjunto. A Figura 6 ilustra estas duas situações.

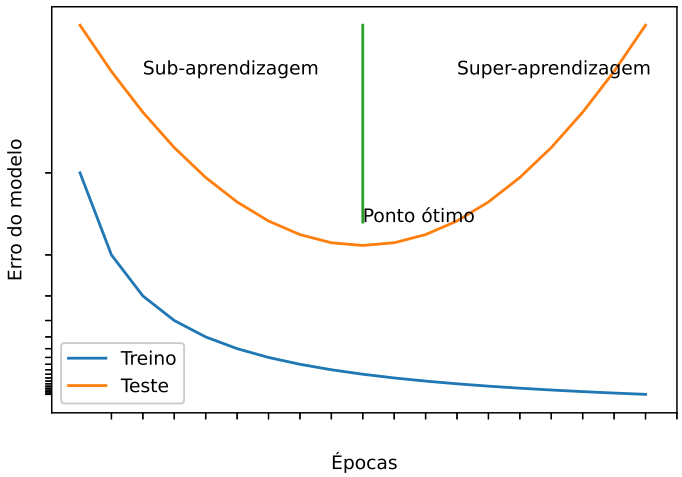


Figura 6 – Gráfico representando a diferença entre sub e super-aprendizagem. A linha azul representa o erro do modelo no conjunto de aprendizagem, e é perceptível que ele só decresce com o passar das épocas. No entanto, para o conjunto de validação e teste (linha laranja), há um ponto ótimo, representado entre a super e a sub-aprendizagem

2.2.2 Descida de Gradiente

Para treinar uma rede e evitar os fenômenos de super e sub-aprendizagem, usa-se o algoritmo de descida de gradiente. Esse algoritmo atualiza os pesos da rede neural baseado no princípio de que o mínimo é alcançado quando o gradiente é nulo. Assim, modificam-se os pesos de acordo com a direção contrária do gradiente, conforme mostra a seguinte equação:

$$w_n^k = w_{n-1}^k - \eta(\nabla J), \tag{2.10}$$

onde w_n^k é o peso na n-ésima iteração da k-ésima camada e η é um hiper-parâmetro chamado de taxa de aprendizagem. É muito importante que o valor da taxa de aprendizagem esteja adequado ao treinamento [Zulkifli 2018]. Valores muito baixos aumentam muito o tempo de convergência e valores muito altos podem fazer o algoritmo divergir, conforme ilustra

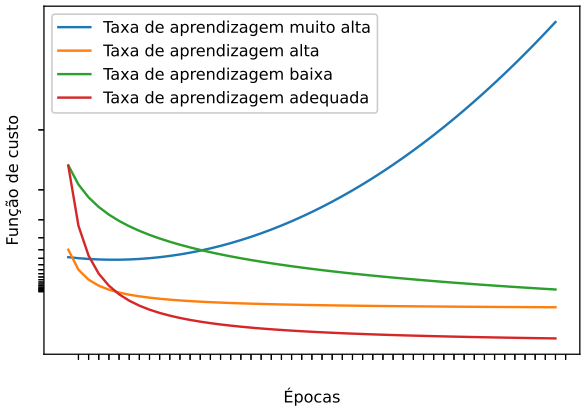


Figura 7 – Ilustração da taxa de aprendizagem. Percebe-se que a taxa de aprendizagem adequada possui o menor valor possível da função de custo e uma convergência rápida. No caso de valores altos, a convergência continua rápida porém o valor atingido é maior, o que diminui a performance do modelo. Se a taxa de aprendizagem for ainda maior, não há convergência. No entanto, valores baixos necessitam de muitas épocas para convergir.

a Figura 7. Uma prática comum é fazer $\eta = 10^{-3}$ [Ballestar e Vilaplana 2020] [Milletari, Navab e Ahmadi 2016] [Myronenko 2018].

Para que o algoritmo convirja mais rapidamente nas primeiras épocas e mais lentamente nas seguintes, atualiza-se a taxa conforme o número de épocas. Assim, garante-se um tempo menor para a convergência do algoritmo de treinamento, mas sem as desvantagens de uma grande taxa. Esses malefícios são um aumento no valor da função de custo, uma menor generalização e um menor desempenho do modelo. Uma forma de atualização é a seguinte [Yu et al. 2018]:

$$\eta_n = \eta_0 \left(1 - \frac{n}{N}\right)^p, \tag{2.11}$$

em que p é um hiper-parâmetro a ser definido e N o número máximo de épocas.

2.2.3 Adam

Adam é um algoritmo de otimização introduzido originalmente em 2014 [Kingma e Ba 2017] e aprimorado diversas vezes desde então. O algoritmo original necessita de três hiper-parâmetros: momentos de primeira e segunda ordem (β_1 e β_2) e a taxa de aprendizagem η. A atualização dos parâmetros w_n é feito por meio da seguinte fórmula:

$$w_n^k = w_{n-1}^k - \eta \frac{m_n}{\sqrt{v_n} + \epsilon}, \tag{2.12}$$

em que ϵ é um número pequeno utilizado para evitar divisões por 0. Por sua vez, m_n e v_n são calculados de forma iterativa através das seguintes equações:

$$m_n = \frac{\beta_1 m_{n-1} + (1 - \beta_1)\nabla J}{1 - \beta_1^n} \tag{2.13}$$

e

$$v_n = \frac{\beta_2 v_{n-1} + (1-\beta_2)(||\nabla J||)^2}{1-\beta_2^n}. \tag{2.14}$$

A inicialização de m_n e v_n são nulas, ou seja, $m_0 = v_0 = 0$. A principal vantagem do Adam é sua rápida convergência [Kingma e Ba 2017] sem comprometer a eficiência do modelo.

2.2.4 RmsProp

Outro algoritmo de otimização muito usado é o RmsProp (*Root Mean Square Propagation*) [Graves 2014]. Esse otimizador funciona de uma forma parecida com o algoritmo Adam. Para a i-ésima época, as equações para atualização da matriz de pesos w_t na rede são:

$$n_i = \rho n_{i-1} + (1-\rho)||\nabla J(w_t)||^2, \tag{2.15}$$

$$g_i = \rho g_{i-1} + (1-\rho)||\nabla J(w_t)||, \tag{2.16}$$

$$b_i = \alpha b_{i-1} + \alpha \frac{\nabla J(w_t)}{\sqrt{n_i + g_i^2 + \epsilon}}, \tag{2.17}$$

$$W_i = W_{i-1} + b_i, \tag{2.18}$$

em que α, ρ e $\epsilon \in (0,1)$ são hiper-parâmetros do algoritmo. Os valores usados por Graves [Graves 2014], por exemplo, foram: $\rho = 0,95$; $\alpha = 0,9$; $\epsilon = 10^{-4}$.

2.3 *Data augmentation*

Uma técnica muito comum usada para melhorar a performance de ANNs é o *data augmentation*. Ela consiste no uso de transformações aleatórias nos dados para aumentar o conjunto de treinamento e permitir uma maior generalização do modelo. No caso do tratamento de imagens, as transformações mais comuns são:

- Rotação: Rotaciona a imagem num ângulo θ, entre $-\theta_{max}$ e θ_{max}, de forma uniforme: $\theta \sim \mathcal{U}(-\theta_{max}; \theta_{max})$.
- Escala: Modifica o tamanho da imagem através de uma multiplicação do tamanho dela por um parâmetro $\beta_{max} \approx 0$ que tem uma distribuição uniforme $\beta \sim \mathcal{U}(1 - \beta_{max}; 1 + \beta_{max})$.
- Ajuste de brilho: Modifica o valor de cada pixel da imagem através de uma multiplicação desse valor por um parâmetro $\gamma_{max} \approx 0$ que tem uma distribuição uniforme $\gamma \sim \mathcal{U}(1 - \gamma_{max}; 1 + \gamma_{max})$.

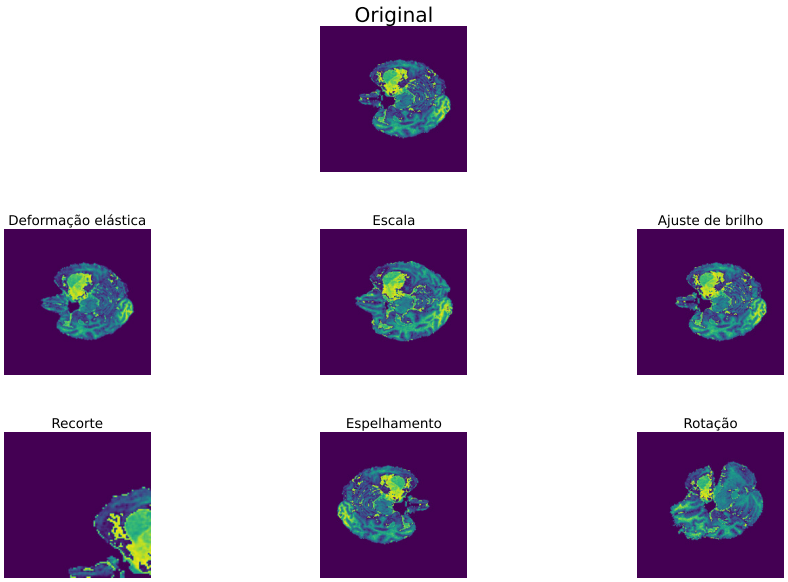


Figura 8 – No topo da figura temos a imagem original e abaixo os exemplos de *data augmentation* (da esquerda para a direita, de cima para baixo): deformação elástica, escala, ajuste de brilho, recorte, espelhamento e rotação.

- Deformação elástica: Um parâmetro α que varia uniformemente e um filtro gaussiano $G(\sigma)$ são aplicados na imagem I de tamanho $M \times N$. A imagem I' é obtida por meio da equação [Castro, Cardoso e Pereira 2018]:

$$I'(j + \Delta_x(j,k), k + \Delta_y(j,k)) = I(j,k), \tag{2.19}$$

  em que $\Delta_x = G(\sigma).\alpha.Rand(N,M)$, $\Delta_y = G(\sigma).\alpha.Rand(N,M)$ e $Rand(N,M)$ é uma função que escolhe aleatoriamente um número entre M e N seguindo a distribuição gaussiana.

- Recorte da imagem: Ao invés de usar a imagem inteira como entrada da rede, apenas uma parte dessa imagem é de fato colocada na rede. Um redimensionamento é feito a fim de garantir que todas as imagens do lote tenham o mesmo tamanho. Por exemplo, se uma imagem tem tamanho 512×512, pode-se usar apenas as primeiras 256 linhas e as primeiras 256 colunas da imagem, criando assim uma imagem de 256×256. Depois, essa nova imagem é redimensionada para ter o tamanho da original, 512×512.

- Espelhamento: A imagem é espelhada em um dos eixos: x, y ou z, no caso de uma imagem tridimensional.

Exemplos de cada uma das transformações numa MRI são apresentados na Figura 8.

De acordo com Cirillo, Abramian e Eklund [Cirillo, Abramian e Eklund 2021], todas as transformações melhoram a eficiência do modelo. É importante ressaltar que as

transformações são introduzidas de forma aleatória, ou seja, há uma probabilidade de 50% de que uma imagem do conjunto de aprendizagem tenha passado por pelo menos uma dessas transformações antes de ser processada pela rede neural nesse trabalho..

2.4 Redes neurais convolutivas

Na segmentação semântica, as redes neurais mais usadas são as Redes Neurais Convolutivas (Convolutional Neural Networks - CNNs, em inglês) [Rawat e Wang 2017]. Elas são definidas como redes neurais que usam operadores de convolução em pelo menos uma das suas camadas, ao invés de fazer uma multiplicação de matrizes. Dessa forma, essas redes geralmente dispõem de dois tipos de operadores não encontrados em outras redes: a convolução e o *pooling*, descritos a seguir.

2.4.1 Operador *pooling*

O comum nas CNNs é diminuir o tamanho da imagem por meio dos operadores de *pooling*, ilustrados na Figura 9. O processo de *pooling* é feito em duas etapas. Primeiramente, divide-se a imagem em imagens menores de mesmo tamanho, chamadas de janelas. Em seguida, aplica-se uma operação para reduzir cada janela em um único valor. Essa operação pode ser a média de todos os valores da janela (*average pooling*) , ou o cálculo do valor máximo dos pixels na janela (*maximum pooling*).

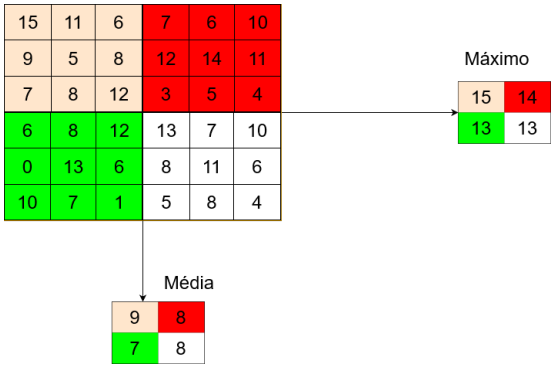


Figura 9 – Exemplo da diferença entre os *poolings* com uma janela 3×3.

2.4.2 Convolução

O operador principal de uma rede convolutiva é evidentemente a convolução. A convolução de uma imagem I de tamanho $M \times N$ é definida da seguinte forma para um ponto $(x, y) \in I$:

$$(W * I)(x, y) = \sum_{i=0}^{m} \sum_{j=0}^{n} W[i, j] I[x + i, y + j], \tag{2.20}$$

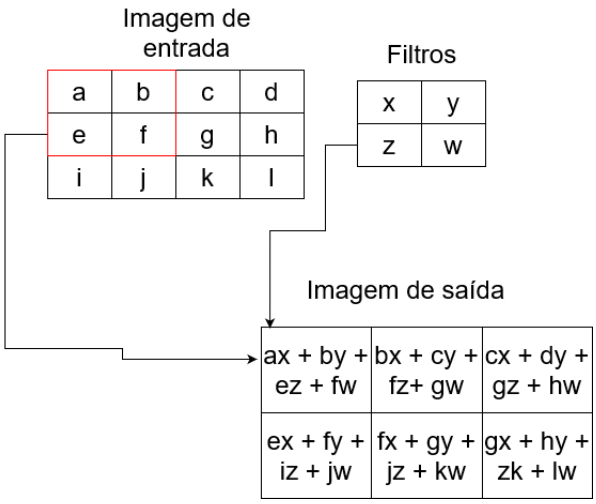


Figura 10 – Exemplo de uma convolução. O retângulo vermelho representa a parte da imagem da entrada usada para calcular o primeiro pixel da imagem de saída.

em que W de tamanho $m \times n$ é uma matriz de filtros (ou *kernel*, em inglês), análogo aos pesos de uma ANN. A Figura 10 ilustra um exemplo de convolução para uma imagem de tamanho 4×4 e um filtro 2×2.

Pelo exemplo apresentado na Figura 10, percebe-se que a matriz de filtros se move sob os pixels da imagem. O número de pixels movidos por vez é chamado de passo (*stride* em inglês), como visto na Figura 11. Naturalmente, quanto maior o passo, menor o tamanho da saída pois a operação é calculada em menos pixels. Assim, variar esse parâmetro pode ser uma alternativa quando se quer substituir a operação de *pooling*.

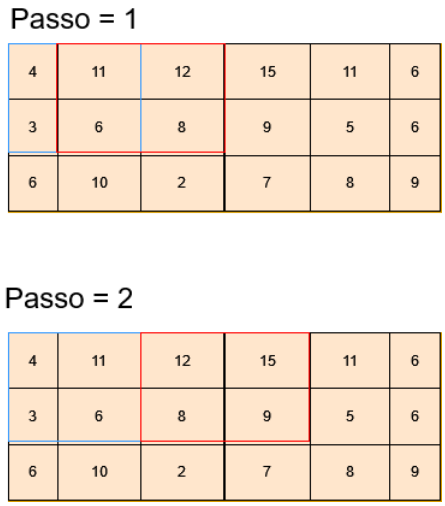


Figura 11 – Esquemático de um passo para uma matriz de filtros 2×2. A janela inicia-se na posição azul e vai para a posição vermelha.

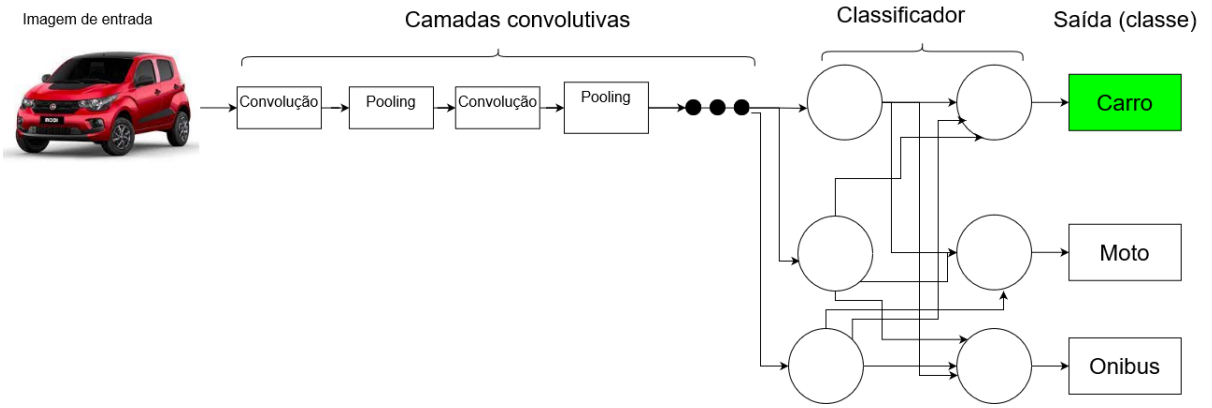


Figura 12 – Estrutura básica de uma CNN. A imagem de entrada, um carro, passa por uma série de camadas convolutivas, composta por uma convolução e um *pooling*, extraindo características da imagem. Essa rede termina com um classificador, que serve para decidir qual é a classe dentre três possíveis: carro, ônibus e moto.

2.4.3 Estrutura das CNNs

Para entender completamente uma CNN, é necessário compreender não somente os operadores, explicados nas subseções anteriores, mas também sua estrutura. A estrutura clássica desse tipo de rede é apresentada na Figura 12.

Nessas redes, as camadas convolutivas servem para extrair características das imagens [Rawat e Wang 2017], enquanto que o operador *pooling* é responsável por diminuir o tamanho da imagem. Esses dois tipos de operadores juntos, convolução e *pooling*, formam um bloco. O comum é a imagem passar por vários blocos e chegar a uma última camada densa, que funciona como um classificador da imagem de entrada. Essa ideia é a base de muitas arquiteturas, como a FCN (*Fully Convolutional Network*) [Long, Shelhamer e Darrell 2015], a RCNN (*Recursive Neural Network*) [Zhang et al. 2020] e a DCNN (*Deep Convolution Neural Network*) [Acharya et al. 2017]. Apesar de a estrutura básica da CNN ter trazido um grande avanço ao processamento de imagens, trabalhos mais recentes alteram essa estrutura [Khan et al. 2020]. Um exemplo de alteração é o uso de uma convolução de passo 2 e a eliminação do operador *pooling*.

2.5 Degradações

Um problema sempre presente em qualquer sistema de engenharia são as degradações. Definimos a degradação como qualquer modificação no sinal que possa reduzir a sua qualidade. Essas degradações podem ocorrer em qualquer etapa: extração, codificação, transmissão ou armazenamento do sinal ou imagem. A seguir, descrevemos alguns tipos de degradações.

- Ruído: De acordo com Gonzalez [Gonzalez e Woods 2008], a imagem com ruído pode

ser modelada da seguinte forma:

$$I(x, y) = I_0(x, y) + n(x, y), \tag{2.21}$$

em que $I_0(x, y)$ é a imagem original, x, y representam as coordenadas, em pixels e $n(x, y)$ é o ruído. Por ter natureza aleatória, o ruído $n(x, y)$ pode ser modelado por meio das funções de densidade de probabilidade $p(z)$, em que z é uma variável aleatória. Os tipos mais comuns de ruído para imagens são:

- Impulso (sal e pimenta): Esse ruído aparece geralmente devido a falhas de sensores na hora de capturar o sinal eletromagnético [Ali 2016]. Sua função de distribuição de probabilidade é:

$$p(z) = \begin{cases} P_a & \text{se } z = a, \\ P_b & \text{se } z = b, \\ 0 & \text{caso contrário.} \end{cases}$$

  No caso de uma imagem de 8 bits comum, temos $a = 0$ e $b = 255$, ou seja, alguns pixels se tornam pretos (a pimenta) e outros brancos (o sal).

- Gaussiano: O ruído Gaussiano possui a $p(z)$ como uma função gaussiana. A seguinte equação mostra a sua função de densidade de probabilidade :

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma}}, \tag{2.22}$$

  em que μ e σ são, respectivamente, a média e o desvio-padrão do ruído.

- *Spike*: Devido ao ruído eletromagnético de alguns sensores [Graves e Donald 2013], na hora de codificar a imagem alguns pontos podem ter uma intensidade muito maior ou menor que a de outros. Esse fenômeno é chamado de *spike*. Essa degradação é caracterizada por algumas linhas pretas na imagem.

- Fantasma: Essa degradação aparece durante a codificação da imagem e é causado por uma frequência de amostragem do sinal MRI insuficiente, caracterizando o fenômeno do *aliasing* [Geissler et al. 2014]. Essa degradação é percebida através do aparecimento de cópias dos objetos da imagem original.

- Degradação por Movimento: Na aquisição de MRIs, os sinais eletromagnéticos detectados são codificados espacialmente por meio de gradientes das imagens. Esses dados codificados formam o chamado k-espaço. Algoritmos de reconstrução, como a transformada inversa de Fourier, são aplicados nesse k-espaço para obter a imagem final. Quando o paciente se move, fazendo uma rotação ou translação, de forma voluntária ou involuntária, perturbações são produzidas tanto na amplitude quanto na fase do sinal eletromagnético. Essa perturbação se propaga nos algoritmos de

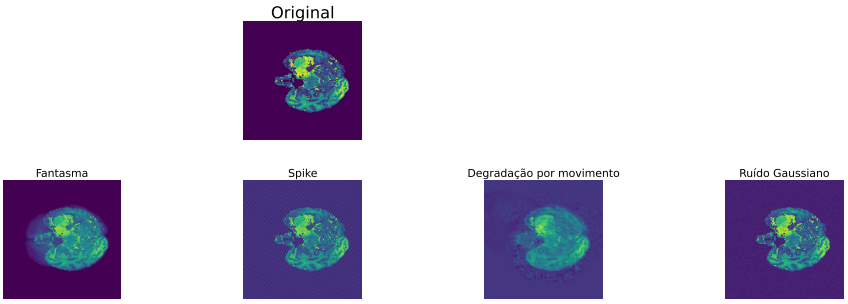


Figura 13 – Exemplos de degradação (da esquerda para a direita, de cima para baixo): fantasma, *spike*, degradação por movimento e ruído gaussiano.

reconstrução. Como resultado, a imagem final é degradada [Godenschweger et al. 2016]. Exemplos de cada uma dos tipos de degradação são apresentados na Figura 13.

2.5.1 Métricas de fidelidade

As métricas de fidelidade são equações que medem o quanto uma imagem de teste, provavelmente com degradações, se difere da imagem original. Os exemplos mais conhecidos são o PSNR (*Peak Signal-to-Noise Ratio* ou relação pico sinal-ruído), o SNR (*Signal-to-Noise Ratio* ou relação sinal-ruído), o MSE (*Mean Squared Error* ou média do erro quadrático), entre outros. Nesse trabalho escolhemos usar o PSNR como métrica de fidelidade. Para entendermos essa métrica, é necessário conhecer o MSE entre uma imagem original (de referência) e a imagem de teste correspondente:

$$MSE = \sqrt{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}[I_0(i,j) - I(i,j)]^2}, \tag{2.23}$$

em que I_0 é a imagem original e I a imagem de teste, ambas de tamanho $m \times n$. O PSNR é calculado por meio da seguinte expressão [Kaur e Singh 2011]:

$$PSNR = 20 \log \frac{\max(I)}{MSE}, \tag{2.24}$$

em que $max(.)$ é o máximo valor possível que um pixel da imagem pode assumir. Por exemplo, no caso de uma imagem codificada em 8 bits, esse valor é igual a 255.

Depois de definida a codificação da imagem, o valor de $\max(I)$ é constante. Assim, o PSNR é uma função decrescente e não-linear do MSE: $MSE \to \infty$ implica $PSNR \to -\infty$ e $MSE \to 0$ implica $PSNR \to \infty$. Logo, quanto maior o valor do PSNR, menos degradante é o sistema para o sinal.

3 Ferramentas

Nesse capítulo, apresentam-se todas as ferramentas específicas desse trabalho: as arquiteturas CNNs, o banco de dados e as métricas de desempenho.

3.1 Métricas de desempenho

Uma parte importante do aprendizado de máquina é a aferição do seu desempenho. Naturalmente, há a necessidade de quantificá-lo, de forma a saber qual modelo se comporta melhor para um determinado problema. As métricas de desempenho mais comuns são [Tiu 2019]:

- Precisão pixel a pixel;
- Interseção sobre união (*Intersection over Union* -IoU, em inglês);
- Dice;
- Distância de Hausdorff.

Todas as métricas, com exceção da Distância de Hausdorff, possuem propriedades semelhantes, valores entre 0 e 1 (0 o pior caso e 1 o ideal) que geralmente são expressos em porcentagens para melhor entendimento. Além disso, os valores dessas métricas podem ser calculados diretamente da matriz de confusão. A matriz de confusão M é definida como uma matriz de tamanho $n \times n$, sendo n o número de classes, cujo elemento m_{ij} é o número de vezes que o modelo previu um pixel como pertencente a uma classe i apesar de esse pixel pertencer à classe j. Nessa definição é interessante notar que a diagonal da matriz representa as predições corretas pois a classe prevista é a mesma da esperada, enquanto que os outros elementos representam previsões errôneas. A distância de Hausdorff, por outro lado, é interpretada de outra forma. Quanto menor a distância de Hausdorff melhor é o modelo, e os valores da distância de Hausdorff variam de 0 a $+\infty$.

3.1.1 Precisão pixel-a-pixel

A precisão pixel-a-pixel, η, mede quantos pixels corretos o modelo prediz num determinado conjunto, em relação ao número total de pixels desse conjunto. Essa métrica é calculada por meio da matriz de confusão utilizando a seguinte equação:

$$\eta = \frac{\sum_{i=1}^{n} m_{ii}}{\sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} m_{ij}}. \tag{3.1}$$

Embora essa métrica pareça adequada para o problema, pois ela indica quantos pixels foram corretamente previstos, ela tem problemas quando os bancos de dados são altamente desbalanceados, ou seja, quando os bancos apresentam uma concentração muito grande de uma das classes. Assim, para o modelo ter uma alta precisão pixel a pixel, basta ele acertar predominantemente apenas a classe que mais tem representante mesmo que erre bastante as outras classes.

3.1.2 Interseção sobre união (*Intersection over Union*-IoU, em inglês)

Como o próprio nome diz, essa métrica tenta relacionar o quanto da área total estabelecida pelo modelo é de fato correta. Sua definição matemática é:

$$\frac{|\hat{V} \cap V|}{|\hat{V} \cup V|}, \tag{3.2}$$

em que $\hat{V}$ e V são, respectivamente, os valores preditos e os verdadeiros de uma imagem. A equação que segue apresenta como obter o IoU por meio da matriz de confusão,para uma i-ésima classe:

$$\frac{m_{ii}}{\sum_{j=1}^{n}(m_{ji} + m_{ij}) - m_{ii}}. \tag{3.3}$$

3.1.3 Dice

Na seção 2.1.1, uma função de custo chamada Função Dice foi apresentada. Sua definição original (equação 2.7) é uma métrica, que é adaptada para ser uma função de custo. De forma semelhante a IoU, essa métrica também pode ser obtida pela matriz de confusão para uma i-ésima classe, utilizando a seguinte equação:

$$2\left(\frac{m_{ii}}{\sum_{j=1}^{n}(m_{ji} + m_{ij})}\right). \tag{3.4}$$

Tanto o Dice quanto a IoU estão relacionadas por meio da seguinte equação:

$$D = 2\left(\frac{IoU}{1 + IoU}\right). \tag{3.5}$$

Vale salientar que há uma função bijetiva que relaciona o Dice ao IoU. Portanto, ao conhecer o valor de uma dessas duas métricas, o valor da segunda é facilmente descoberto, pois basta resolver a equação 3.5 usando o valor conhecido.

3.1.4 Distância de Hausdorff

De acordo com Ribera *et al.* [Ribera et al. 2019], a distância de Hausdorff d_H é uma medida que quantifica o quão dois conjuntos são diferentes, sendo definida pela seguinte equação:

$$d_H(X, Y) = \max\{\sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y}(\inf_{x \in X} d(x, y)\}, \tag{3.6}$$

em que X, Y são dois conjuntos quaisquer, $d(x, y)$ é uma função de distância de dois pontos ou vetores, sup e inf representam o supremo e o ínfimo de um conjunto, respectivamente. No caso desse projeto, tanto o supremo quanto o ínfimo podem ser entendidos como máximo e mínimo, respectivamente.

No caso desse trabalho, $d(x, y)$ é a distância euclidiana ($||x-y||_2 = \sum_i (x_i - y_i)^2, x \in X, y \in Y$), X é o conjunto de pontos de uma imagem predita e Y o conjunto de pontos da imagem desejada. Ambos conjuntos são limitados e discretos, ou seja, seus elementos possuem um valor máximo e ambos pertencem ao conjunto dos números inteiros. Essas propriedades dos conjuntos permitem simplificar a equação. Além disso, usam-se não apenas uma, mas para avaliar o desempenho do modelo utilizam-se várias imagens ao mesmo tempo. Portanto, é mais interessante analisar a média das distâncias de Hausdorff $\bar{d}_H$:

$$\bar{d}_H(X, Y) = \frac{1}{|X|} \sum_{x \in X} \min_{y \in Y} d(x, y) + \frac{1}{|Y|} \sum_{y \in Y} \min_{x \in X} d(x, y) \tag{3.7}$$

3.2 BRATS

É importante conhecer o banco de dados utilizado nesse trabalho pois, como o modelo é treinado nesse banco, o resultado obtido somente é valido em bancos parecidos. Assim, é necessário ter informações da proporção de classes, do número de classes, do número de imagens, do tamanho das imagens, da forma de aquisição dessas imagens e de sua codificação.

O banco de dados escolhido neste trabalho foi o 2020 Brain Tumor Segmentation (BraTS), um dos poucos bancos disponíveis para a segmentação semântica de gliomas cerebrais. De acordo com Bakas *et al.* [Bakas et al. 2019], os bancos anteriores eram privados e muito diferentes entre si, o que dificulta a comparação dos resultados. Essas diferenças existem por diversos motivos, como por exemplo tipo de tumor de cada paciente, estado da doença e sigilo médico.

3.2.1 Composição

O banco de dados é composto por um conjunto de imagens de 335 pacientes resultando em $3,5$ GB de dados. Para cada paciente, existem 4 MRIs: ativo (T1), pós-contraste ponderada em T1 (T1Gd), ponderada em T2 e recuperação de inversão atenuada por fluido T2 (T2-FLAIR). Além disso, há uma última imagem onde cada pixel representa cada uma das quatro classes possíveis: Fundo da imagem ou cérebro sadio (classe 0); *Necrotic and non-enhancing tumor core* (NCR/NET — label 1); *Peritumoral edema* (ED — classe 2); *GD-enhancing tumor* (ET — classe 4).

Todas as imagens do banco BRATS são tridimensionais, com $240 \times 240 \times 150$ pixels. A Figura 14 mostra um exemplo de uma das imagens do BRATS. Percebe-se que a

visualização do tumor fica prejudicada quando a imagem é vista dessa forma. Uma forma de contornar isso é visualizar os cortes da imagem. Por exemplo, podemos fatiar a imagem em 150 camadas no eixo z e selecionar diferentes cortes. A Figura 15 mostra o tumor em um desses cortes.

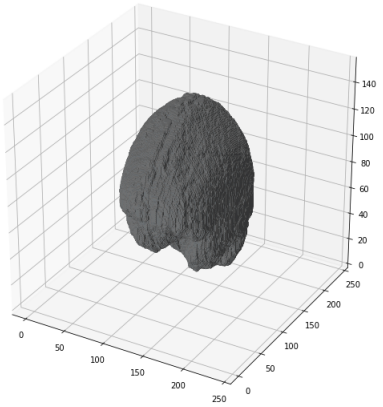


Figura 14 – Imagem de um cérebro do banco de dados representada em 3D.

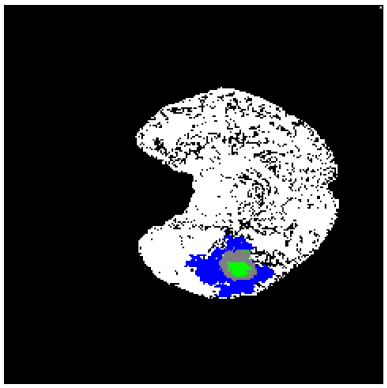


Figura 15 – Classes do banco de dados. As partes verdes, cinzas e azuis representam o NET (classe 1), o ED (classe 2) e o ET (classe 4), respectivamente. Os pixels restantes da imagem pertencem ao fundo da imagem (classe 0).

3.2.2 Distribuição estatística

No caso de uma segmentação semântica, é importante conhecer a proporção de pixels de cada classe, uma vez que isso influencia muito o desempenho do modelo. Assim, analisando as imagens do BRATS de cada paciente e contando o numero de pixels em cada classe obtivemos a proporção de dados apresentada na Tabela 1. Observe que quase todos os pixels do banco pertencem à mesma classe, que é justamente a classe que está

Classe	Proporção (%)
0	98,88
1	0,24
2	0,64
4	0,22

Tabela 1 – Proporção dos pixels divididos em classes. Para facilitar a leitura, o resultado de cada classe é apresentado em % do número total de pixels.

fora do escopo deste projeto, pois o nosso objetivo é detectar e classificar tumores e não o fundo da imagem.

3.2.3 Bancos altamente desbalanceados

Como visto na seção 3.2.2, há um alto desbalanceamento das classes do BRATS. Isso é um problema, pois o modelo tende a classificar tudo como a classe estatisticamente majoritária, ou seja, o fundo da imagem. Se isso acontecer, o modelo tem uma precisão pixel a pixel bem elevada, acima de 90%. Contudo, o mesmo modelo também erra todas as outras classes e não possui utilidade na prática, uma vez que o objetivo desse modelo é mostrar ao médico onde fica o tumor no cérebro. Há algumas formas de resolver o problema de bancos altamente desbalanceados:

- Mudança de amostragem: Essa técnica consiste em aumentar artificialmente o número de representantes de uma classe minoritária, o tumor, ou diminuir a de uma classe majoritária, o fundo da imagem [Tang et al. 2009].
- Uso de pesos: Uma outra forma de melhorar a performance é alterando a função de custo, por meio da introdução de pesos nessas funções. Esses pesos devem ser introduzidos de forma que quanto maior a proporção de uma classe, menor é o seu peso. Assim, o modelo é mais penalizado quando o erro é feito nas classes minoritárias [Naceur et al. 2019].
- Mudança na função de custo: Ao invés de simplesmente adaptar a função com pesos, outra solução é a mudança da função. Por exemplo, Kervadec *et al.* [Kervadec et al. 2019] propuseram uma função auxiliar que leva em consideração a fronteira das regiões de cada classe.

3.3 Arquiteturas CNN utilizadas

Essa seção se refere às arquiteturas CNN estudadas no trabalho. Essas arquiteturas são adequadas ao problema pois na literatura elas foram implementadas no BRATS 2020

para a realização de segmentação semântica e obtiveram resultados promissores, acima de 80% no coeficiente Dice.

3.3.1 3D U-net

A U-net é uma das arquiteturas mais populares para segmentação semântica. Essa arquitetura possui esse nome devido ao seu formato em "U", conforme mostrado na Figura 16. O bloco azul representa a imagem 3D de entrada. Conforme ela passa pela rede, seu tamanho é alterado, como ilustrado pelo esquema. A estratégia da arquitetura consiste em usar blocos de convolução para extrair características da imagem e usar o operador *pooling* para reduzir o tamanho da imagem, de forma semelhante à outras CNNs, descrito na seção 2.4.3. Dessa forma, as primeiras camadas mapeiam características mais gerais, enquanto as camadas seguintes se concentram em características mais específicas da imagem. Vários estudos mostraram a sua eficácia em imagens médicas: [Myronenko 2018], [Ballestar e Vilaplana 2020] e [Henry et al. 2020]. Portanto, é interessante experimentar sua aplicação e comparar com outras propostas.

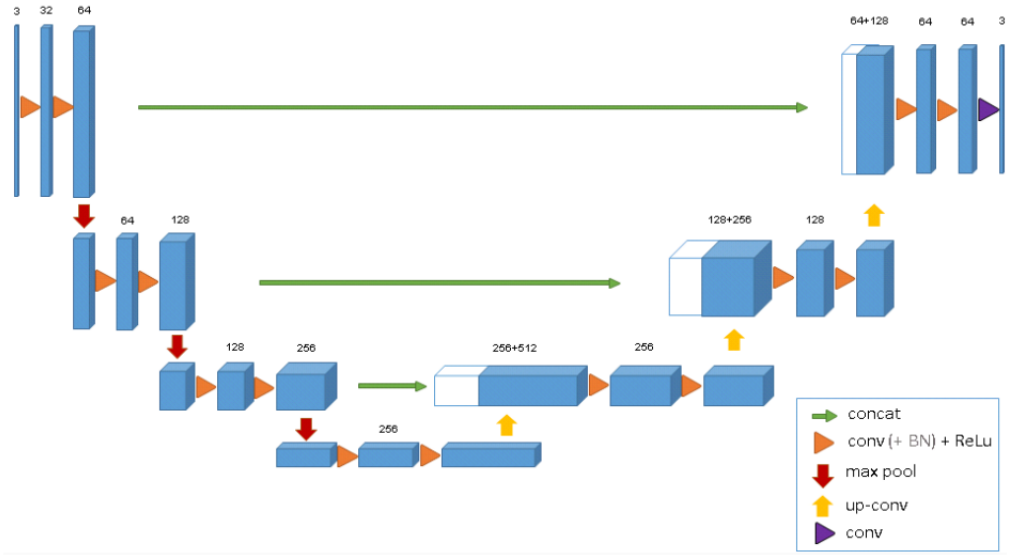


Figura 16 – Esquemático padrão da U-net [Çiçek et al. 2016].

3.3.2 Residual U-net

Um problema comum com redes profundas é a dissipação dos gradientes (*vanishing gradient*, em inglês). Os otimizadores mais populares requerem calcular o gradiente de trás para frente, atualizando sempre os pesos das camadas mais profundas até chegar nas primeiras camadas da rede. O problema dessa forma de cálculo é que, pela regra da cadeia, o gradiente das primeiras camadas é um produto dos gradientes das camadas mais profundas. Assim, para uma rede suficientemente grande, o valor do gradiente das

primeiras camadas tende a ser muito pequeno, pois será um produto de outros gradientes também pequenos. Isso torna o aprendizado da rede muito lento.

A Resnet [He et al. 2015] busca resolver este problema por meio de atalhos que permitem a adição de valores ao gradiente. A cada duas camadas, o valor da entrada da camada é adicionado à saída, como mostra a Figura 17. Nessa rede, a convolução é mantida, porém o operador *pooling* é retirado. A diminuição da imagem normalmente feita pelo *pooling* é substutuída por uma outra convolução de passo 2 no início do primeiro bloco. A Residual U-net, proposta por Ballestar e Vilaplana [Ballestar e Vilaplana 2020], leva em consideração os atalhos da Resnet. Assim, ao invés de usar o bloco padrão (convolução + convolução + maxpool), implementa-se o bloco da Resnet apresentado na Figura 17.

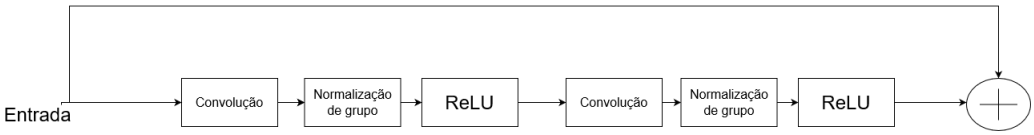


Figura 17 – Representação do atalho. Percebe-se que há uma adição entre a entrada e a saída do bloco.

3.3.3 DMFnet

Uma arquitetura interessante usada em desafios de aprendizado de máquina é a *Dilated Multi-Fiber Net* (DMFNet) [Chen et al. 2019]. Diferentemente das demais, essa arquitetura foi pensada de forma a otimizar o tempo de resposta do modelo. Sua proposta original garante uma velocidade maior que a U-net, pois a arquitetura é consideravelmente menor. A Figura 18 ilustra o esquemático dessa arquitetura.

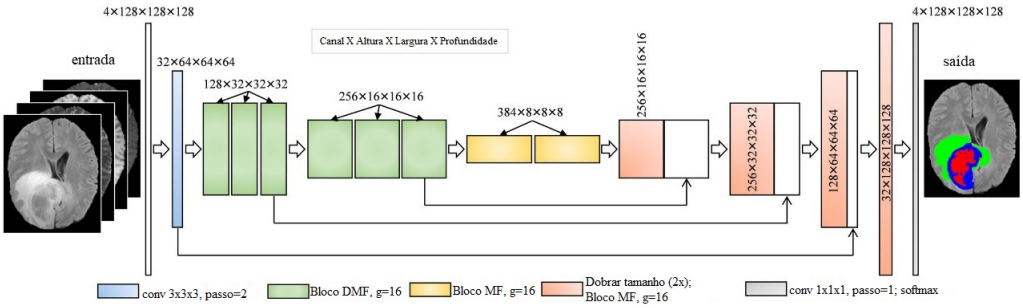


Figura 18 – Esquema da DMFnet [Chen et al. 2019]

Os blocos dessa arquitetura são mostrados na Figura 19. Esse blocos são construídos por meio de fibras, que são definidas na arquitetura como múltiplos blocos da Resnet separados entre si. Além disso, esses blocos diferem do bloco usado na Resnet por possuírem as seguintes características:

- Agrupamento de canais: Em termos computacionais, uma convolução é custosa. Logo, uma estratégia do modelo é reduzir o número de parâmetros da convolução por meio de g blocos de Resnet (as fibras), processados paralelamente. Consequentemente, o tempo de execução é reduzido.

- Multiplexadores: Como o processamento é dividido entre grupos, é necessário fazer uma troca de informação entre essas fibras. O multiplexador, usando parâmetros próprios que são são otimizados assim como o restante do modelo, tem como propósito regular essa troca de troca de informação entre essas fibras. A junção de fibras com multiplexadores define o bloco de Multi-Fibras (MF), esquematizado na Figura 19(c).

- Convolução dilatada: As últimas fibras do modelo usam a convolução dilatada no lugar da comum. A diferença entre uma convolução comum e uma dilatada é que a segunda usa um parâmetro chamado índice de dilatação d, que permite extrair informações de pixels mais distantes e ter uma melhor visão das características gerais da imagem em detrimento de características mais específicas de objetos próximos [Chen et al. 2019]. A Figura 19(e) mostra a diferença desses dois tipos de convolução, de $d = 1$ (convolução convencional) para $d = 2$ (convolução dilatada). Ao usar as convoluções dilatadas com os multiplexadores, define-se o bloco de Multi-Fibras Dilatado, ilustrado na Figura 19(d).

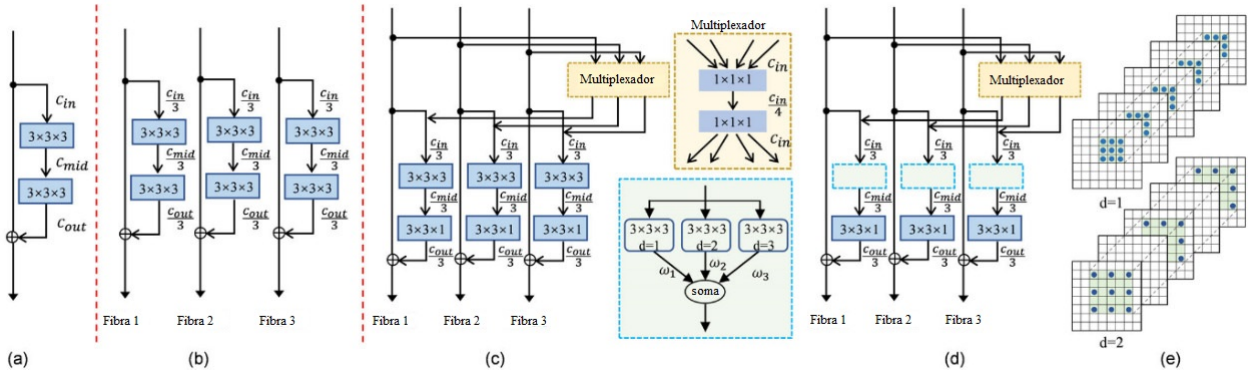


Figura 19 – Blocos propostos por Chen *et al.* [Chen et al. 2019]: (a) Esquema padrão do bloco uma Resnet, inspiração do modelo. (b) As fibras, definidas como múltiplos blocos da Resnet separados entre si. (c) Uso de multiplexadores e fibras para a construção do bloco Multi-Fibras (*Multi-Fiber* - MF, em inglês). (d) O bloco de Multi-Fibras Dilatado (*Dilated Multi-Fiber* - DMF, em inglês). (e) Esquema de uma convolução dilatada. Quando $d = 1$, a matriz de filtros da convolução faz o cálculo apenas com os pixels vizinhos (uma distância de um pixel) da coordenada x, y que é computada. Já quando $d = 2$, não se usam os pixels vizinhos e pixels que estejam exatamente a dois pixels da coordenada x, y.

3.3.4 V-net

A Vnet foi especialmente desenhada para imagens médicas, como mostra seu estudo original [Milletari, Navab e Ahmadi 2016]. Além disso, Ballestar e Vilaplana [Ballestar e Vilaplana 2020] também estudaram sua eficácia em tumores cerebrais e o resultado foi promissor. Por isso, decidimos testar essa arquitetura nesse trabalho. Conforme pode ser observado no seu esquemático apresentado na Figura 20, essa arquitetura é uma variação da U-net padrão.

Na V-net, a parte esquerda da rede funciona como um compressor, reduzindo a resolução da imagem. A parte da direita faz o oposto: aumentando a resolução. A redução da imagem é feita por meio de um processo de sub-amostragem e o aumento, por um processo de super-amostragem. Tanto a super-amostragem quanto a sub-amostragens são feitas por meio de uma convolução de passo 2, ao invés de usar a operação de *pooling*. Vale lembra que esse operador consome muita memória. Além disso, o modelo pode ser visualizado como um conjunto de diferentes estágios. Cada estágio possui três convoluções com tamanhos de matrizes de filtros diferentes, sendo duas de $5 \times 5 \times 5$ e uma de $2 \times 2 \times 2$. Essa última matriz de filtros é responsável por mudar o tamanho da imagem, pois possui um passo igual a 2. Por fim, usa-se uma última camada de neurônios com a operação $softmax$ para prever a localização dos tumores cerebrais.

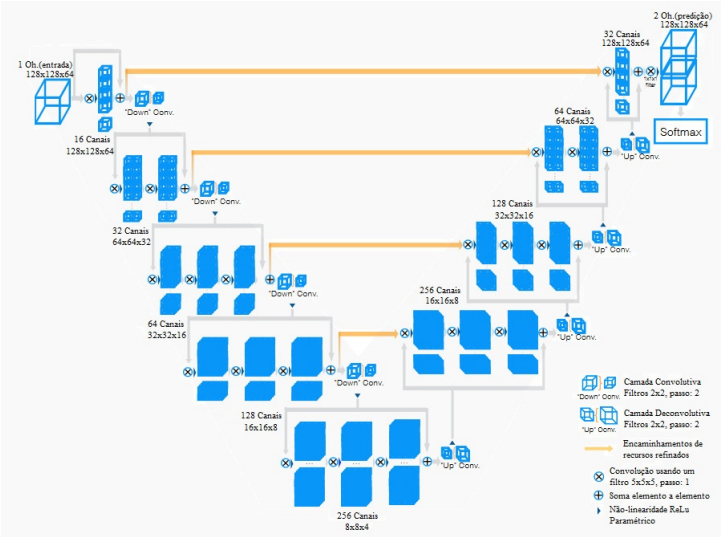


Figura 20 – Esquemático padrão da V-net [Milletari, Navab e Ahmadi 2016].

4 Resultados experimentais

Esse capítulo apresenta os resultados dos experimentos mais relevantes desse trabalho. No entanto, antes de apresentar esses resultados, é importante entender as circunstâncias dos experimentos: hiper-parâmetros, técnicas de *data augmentation* utilizadas, otimizadores testados e função de custo.

4.1 Detalhes do treinamento

Os experimentos foram todos feitos no ambiente colab Pro da Google, com uma GPU Nvidia de 16 GB de RAM. O pré-processamento foi feito por meio da biblioteca numpy, a simulação das degradações por meio da biblioteca torchIO [Pérez-García, Sparks e Ourselin 2021], o processamento por meio da biblioteca pytorch e a exibição dos dados por meio das bibliotecas tensorboard e matplotlib. Além disso, os hiper-parâmetros de todos os treinamentos foram fixados conforme a Tabela 2. O tamanho de lote (*batch size*) foi menor que o usado na literatura devido às limitações da GPU usada.

Como visto na Seção 3.2.1, as imagens são tridimensionais com um tamanho elevado. Por isso, decidiu-se dividir cada uma delas em cubos menores, que foram usados no treinamento ao invés da imagem inteira. Cada cubo possui uma aresta de 128 pixels. Além disso, caso só houver apenas fundo no cubo, aquela área não é levada em consideração na hora do treinamento. Dessa forma, apenas cubos que contêm tumor são usados no treino. Duas técnicas de *data augmentation* também foram implementadas para aumentar a variabilidade dos cubos: o espelhamento e a transformação elástica, conforme detalhadas na Seção 2.3.

Todos os experimentos usaram funções de custo com pesos: 0,1 para o fundo e 1 para o restante. Primeiramente, foram feitos experimentos sem degradações, de forma a comparar outros parâmetros do modelo: arquiteturas, matrizes de confusão e otimizadores. Em seguida, foram realizados experimentos com imagens degradadas.

Hiper-Parâmetro	*Batch Size*	Épocas (máximo)	Divisão do banco (treinamento, validação e teste)	Taxa de aprendizagem
Valor	4	250	60%, 20%, 20%	10^{-3}

Tabela 2 – Tabela de hiper-parâmetros fixos para todos os treinamentos

Otimizadores	U-net		Residual U-net		DMFnet		V-net	
	Dice	Hausdorff	Dice	Hausdorff	Dice	Hausdorff	Dice	Hausdorff
SGD	50,0	12,8	48,4	14,6	19,9	31,4	27,1	26,5
Adam	76,0	9,8	70,8	10,6	73,8	11,1	71,7	9,3
RMSProp	75,3	9,5	72,0	10,2	75,5	11,3	71,2	9,1

Tabela 3 – Médias do coeficiente de Dice e da distância de Hausdorff para as arquiteturas DMFnet, 3D U-net, Residual U-net e V-net utilizando os otimizadores SGD, Adam e RMSProp.

Otimizadores	U-net		Residual U-net		DMFnet		V-net	
	Dice	Hausdorff	Dice	Hausdorff	Dice	Hausdorff	Dice	Hausdorff
SGD	15,1	6,1	28,7	6,4	14,0	9,5	26,5	6,06
Adam	4,6	5,1	6,9	6,3	5,8	5,9	3,2	4,7
RMSProp	5,4	5,0	7,4	6,1	5,9	6,5	4,31	4,7

Tabela 4 – Desvios padrão do coeficiente de Dice e da distância de Hausdorff para as arquiteturas DMFnet, 3D U-net , Residual U-net e V-net utilizando os otimizadores SGD, Adam e RMSProp.

4.2 Otimizadores testados

Nesse trabalho, três otimizadores (SGD, ADAM e RMSProp) foram testados com quatro arquiteturas diferentes (DMFnet,3D U-net,Residual U-net e V-net). A métrica de desempenho foi a média dos coeficientes de Dice e da distância de Hausdorff de cada tumor. A função de custo foi a função de Dice, conforme mencionado nos trabalhos de cada arquitetura [Ballestar e Vilaplana 2020] [Chen et al. 2019]. Assim, a Tabela 3 apresenta todos os resultados obtidos em média para cada uma das arquiteturas no conjunto de validação. Porém, só o valor da média não é capaz de indicar se os resultados foram bons para cada classe, uma vez que pode haver uma grande variação entre os resultados de cada classe. Assim, para ter certeza que os resultados de cada classe foram semelhantes, o desvio-padrão também foi calculado e apresentado na Tabela 4.

Alguns detalhes importantes são perceptíveis tanto para a distância de Hausdorff como para o coeficiente Dice: o otimizador SGD teve um resultado muito abaixo do esperado, enquanto que as arquiteturas tiveram resultados parecidos. O Adam e o RMSProp tiveram resultados semelhantes com um desvio padrão pequeno. Escolheu-se usar o RMSProp para os demais testes.

4.3 Matrizes de confusão

A matriz de confusão, discutida na Seção 3.1, é de suma importância para a análise de uma arquitetura. Ela permite identificar como o modelo se comporta para cada classe, e quais classes são mais confundidas. Nesse trabalho, a matriz permite identificar se o

	Fundo	NCR/NET	ED	ET
Fundo	95,2	0,3	4,1	0,4
NCR/NET	2,4	74,2	14,8	8,6
ED	10,2	5,3	80,6	3,9
ET	6,3	6,8	5,4	81,5

(a) U-NET

	Fundo	NCR/NET	ED	ET
Fundo	97,2	0,4	2	0,4
NCR/NET	3,84	81,1	9,2	5,9
ED	14,1	8,6	74,6	2,6
ET	9	9,5	4,6	76,9

(b) Residual U-NET

	Fundo	NCR/NET	ED	ET
Fundo	97	0,6	1,8	0,6
NCR/NET	3,3	77,4	8,8	10,5
ED	10,5	13,4	69,4	6,7
ET	3,4	6,9	3,7	85,9

(c) DMFnet

	Fundo	NCR/NET	ED	ET
Fundo	96,2	0,6	2,5	0,7
NCR/NET	1,9	65,6	23	9,5
ED	7,6	9,5	78,7	4,2
ET	6,5	8,4	6,6	78,4

(d) V-NET

Figura 21 – Matrizes de confusão para as arquiteturas testadas.

modelo confunde os tipos de tumores ou a parte saudável do cérebro com um tumor. As colunas dessa matriz representam a classe predita, enquanto que as linhas representam a classe esperada.

Para facilitar a visualização, dividiram-se os valores absolutos da matriz pela soma de cada linha, ou seja, o número de pixels para cada classe da saída desejada. Dessa forma, a diagonal principal indica a porcentagem de pixels acertados da classe. Como o banco de dados é muito desbalanceado, a precisão pixel a pixel é aproximadamente igual ao primeiro valor da diagonal principal (m_{11}). O resultado é apresentado na Figura 21.

Percebe-se uma precisão pixel a pixel é maior que 95% para todas as arquiteturas. Além disso, de forma geral, é observado que o modelo confunde mais os tipos de tumores que o cérebro saudável com um tumor. As classes NCR e ED, por exemplo, são bastante confundidas, especialmente nas arquiteturas V-net (23%) e U-net (14,8%). Isso significa que o modelo tem tanta dificuldade de classificar o tumor como de identificar o tumor.

4.4 Resultado para imagens com degradação

Outro objetivo desse trabalho é verificar o quão robustas são as arquiteturas para diferentes tipos e intensidades de degradações. A partir do banco de dados original, geramos um novo conjunto de imagens com 4 tipos de degradação. Foram utilizadas 10 intensidades diferentes para cada tipo de degradação. Os tipos de degradação foram criados artificialmente da seguinte forma:

- Ruído Gaussiano: De acordo com a Equação 2.22, o ruído gaussiano é controlado pelo desvio padrão σ e pela média μ. Nesse trabalho, o desvio padrão do ruído gaussiano é um múltiplo de 0,065 para cada nível n, ou seja, $\sigma = 0,065 * n$. A média do ruído é nula.

- Degradação por Movimento: Na Seção 2.5, descrevemos como a movimentação de

um paciente causa degradações que dependem da rotação e traslação do movimento. Além disso, é possível que o paciente se movimente mais de uma vez durante a extração da imagem. Por isso, além da translação e rotação, consideramos também o número de vezes que o paciente se movimentou. Nas simulações para esse trabalho, há uma rotação r de no máximo 30^o e uma translação t de no máximo 5 mm para cada movimento. Ambas, rotação e translação, seguem uma distribuição uniforme, ou seja, $r \sim \mathcal{U}(-30; 30)$ e $t \sim \mathcal{U}(-5; 5)$. O número de movimentos é igual ao nível da degradação, ou seja, se o nível da degradação é 1, o paciente se movimentou apenas uma vez; se o nível é 2, o paciente se movimentou duas vezes; e assim em diante.

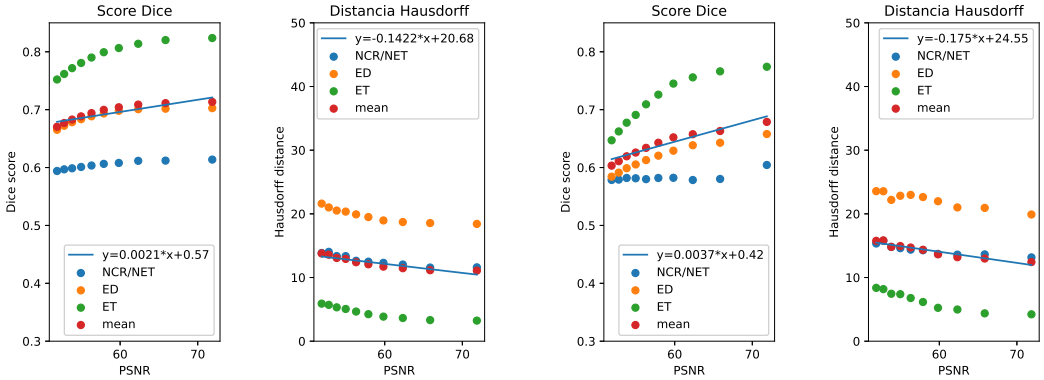
- Fantasma e *spike*: Ambas as degradações dependem do número de fantasmas/*spike*, definida como o número de objetos/pontos aberrantes, como pela intensidade, razão entre o valor dos pontos do fantasma/*spike* e o maior valor da imagem. Nesse trabalho, O número de fantasmas/*spike*s é fixo e igual a 1. Porém, a intensidade i varia como um múltiplo de 0,065$(0,065 * n)$ de forma uniforme,ou seja, $i \sim \mathcal{U}(-0,065 * n, 0; 065 * n)$.

O modelo foi treinado sem degradações, que foram adicionados apenas no teste. Esses exeprimentos permitiram a criação dos gráficos das Figuras 22 e 23, em que se avalia o PSNR do sinal com o Dice e com a distância Hausdorff para cada classe do banco de dados , exceto o fundo, e para cada arquitetura. Por fim, o gráfico apresenta também a média dos resultados de cada classe e uma regressão polinomial de ordem 1 feita com os dados dessa média. Assim, pode-se estimar o quanto as métricas variam em função do PSNR e determinar qual degradação é mais prejudicial ao desempenho do modelo. A degradação de tipo fantasma teve um efeito muito semelhante ao ruído gaussiano, enquanto que a degradação do tipo *spike* teve um efeito praticamente nulo no modelo. Por essas razões, os gráficos de ambas não são apresentadas.

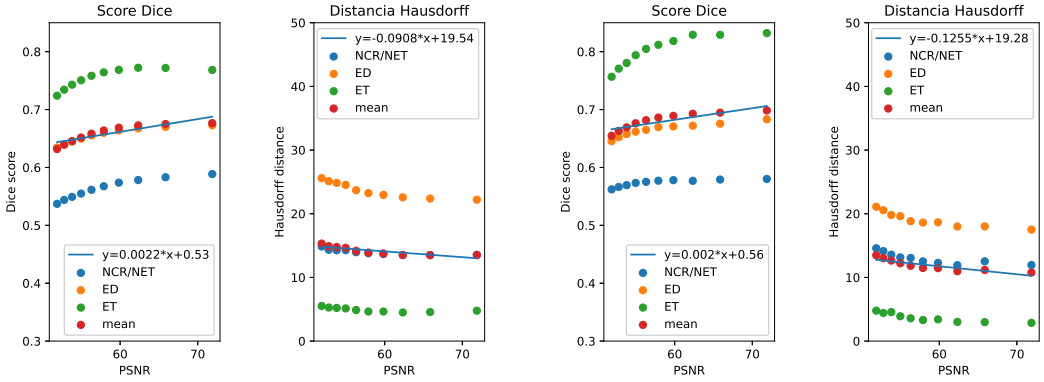
Algumas observações podem ser destacadas pela leitura dos gráficos (Figuras 22 e 23). Essas análises são válidas independentemente da métrica analisada (distância de Hausdorff ou métrica de Dice):

- Em todos os gráficos, há uma regressão linear obtida pelos dados da média das classes, da forma $y = mx + b$. O parâmetro m indica o quanto a métrica (Dice ou Distância de Hausdorff) varia quando há variação do PSNR. Em valores absolutos, o que obteve o menor m foi a V-net, para todas as deformações, e a Residual U-net, o maior deles. Assim, pode-se dizer que a diminuição do PSNR (ou seja, aumento da degradação) afeta menos a V-net e mais a Residual U-net.

- A degradação causada pelo movimento diminui o desempenho do modelo, mais que outros tipos de degradações. No entanto percebe-se que os dados para esse tipo de degradação não formam uma sequência monótona crescente (no caso do



(a) U-NET (b) Residual U-NET



(c) DMFnet (d) V-NET

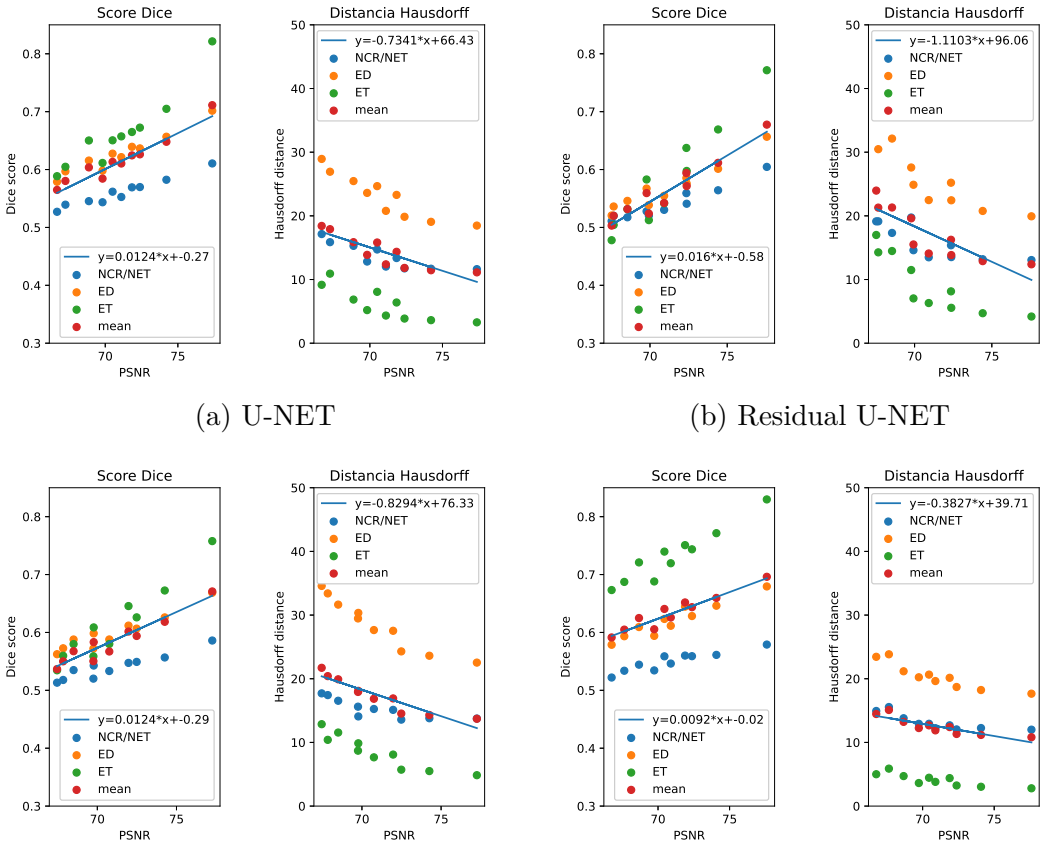
Figura 22 – Gráficos das métricas desempenho Dice e Hausdorff para várias intensidades de ruído gaussiano (e, consequentemente, vários valores de PSNR) para as quatro arquiteturas testadas.

Dice) ou decrescente (no caso da distância de Hausdorff). O provável motivo dessa inconsistência são as variáveis de translação e rotação, que, por terem natureza aleatória, oscilam entre valores altos e baixos. Assim, mesmo que o número de movimentos aumente de acordo com o nível da degradação, se os valores de translação e/ou rotação forem significativamente menores, ainda assim o modelo consegue um desempenho melhor.

- Em termos de desempenho, a média de todas as arquiteturas oscilaram entre valores próximos, entre 65% e 70% na métrica Dice. Essa proximidade é esperada visto que na literatura os desempenhos também foram bem próximos [Ballestar e Vilaplana 2020] [Chen et al. 2019].

4.5 Imagens

Embora as métricas deem uma boa noção de como o sistema funciona, elas não permitem uma análise mais qualitativa e específica. Por isso, é importante também analisar



(a) U-NET (b) Residual U-NET

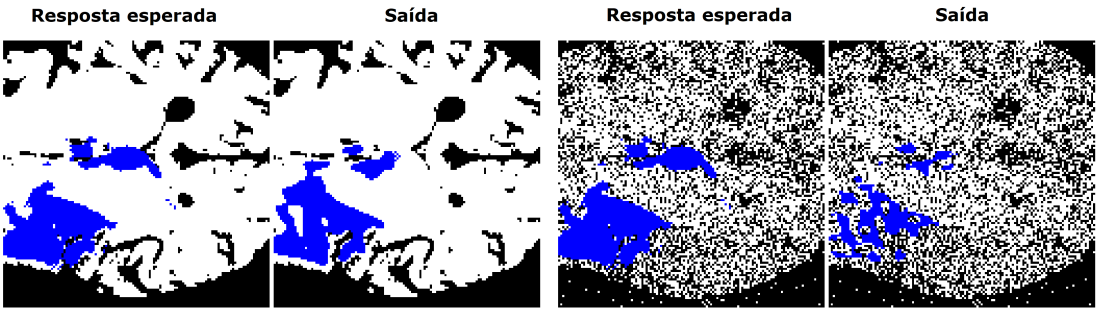
(c) DMFnet (d) V-NET

Figura 23 – Gráficos das métricas desempenho Dice e Hausdorff para várias intensidades de degradações por movimento (e, consequentemente, vários valores de PSNR) para as quatro arquiteturas testadas.

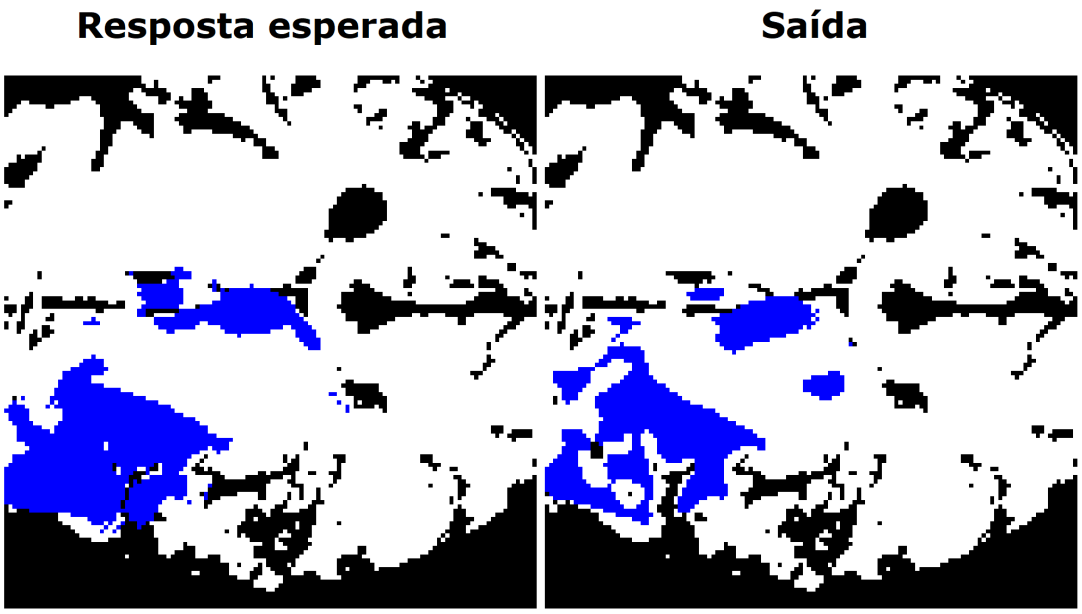
algumas imagens de saída e compará-las com o esperado. Sendo assim, nas Figuras 24, 25, 26 e 27 apresentamos as imagens das identificações e segmentações das áreas com tumores obtidas com as arquiteturas Residual U-net, DMFnet, V-net e U-net, respectivamente. As figuras mostram as áreas de tumores segmentadas em imagens sem degradações (Figuras 24(a), 25(a), 26(a) e 27(a)) e as áreas segmentadas dos tumores em imagens com degradações de ruído e movimento (Figuras 24(b-c), 25(b-c), 26(b-c) e 27(b-c))

Percebe-se que qualitativamente é visível a diferença. Contudo, mesmo com altas degradações, consegue identificar uma parte significativa do tumor, o que sugere que as redes possuem alta robustez às degradações. As arquiteturas também tem resultados parecidos entre si, o que é esperado dado que a diferença nas métricas é bem pequena, conforme visto na Seção 4.4. Além disso, pelas imagens também é possível observar que quase todos os pixels detectados como tumor são de fato tumores. O modelo falha ao classificar tumores como fundo de imagem, mas o sentido inverso (classificar fundo de imagem como tumor) não é visto.

Um ponto a ser notado é que na prática, a intensidade da degradação não é tão



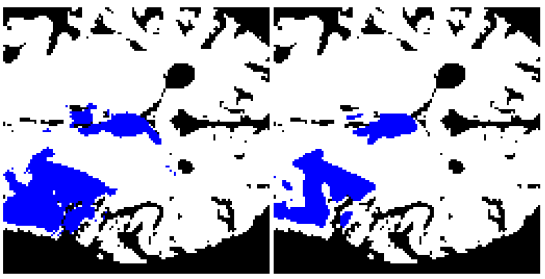
(a) Sem ruído (b) Ruído gaussiano $\sigma = 0,325$



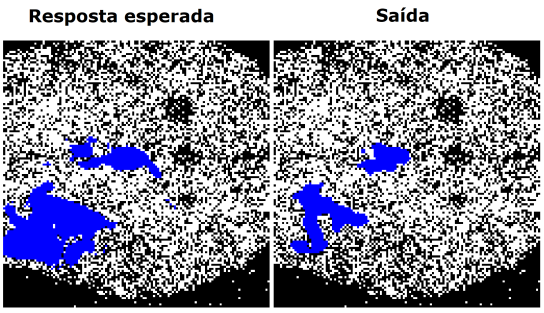
(c) Degradação por movimento de nível 5

Figura 24 – Imagens das identificações e segmentações das áreas com tumores obtidas com a arquitetura Residual U-net para: (a) imagems sem degradações, (b) imagem com ruído e (c) imagem com borrado de movimento.

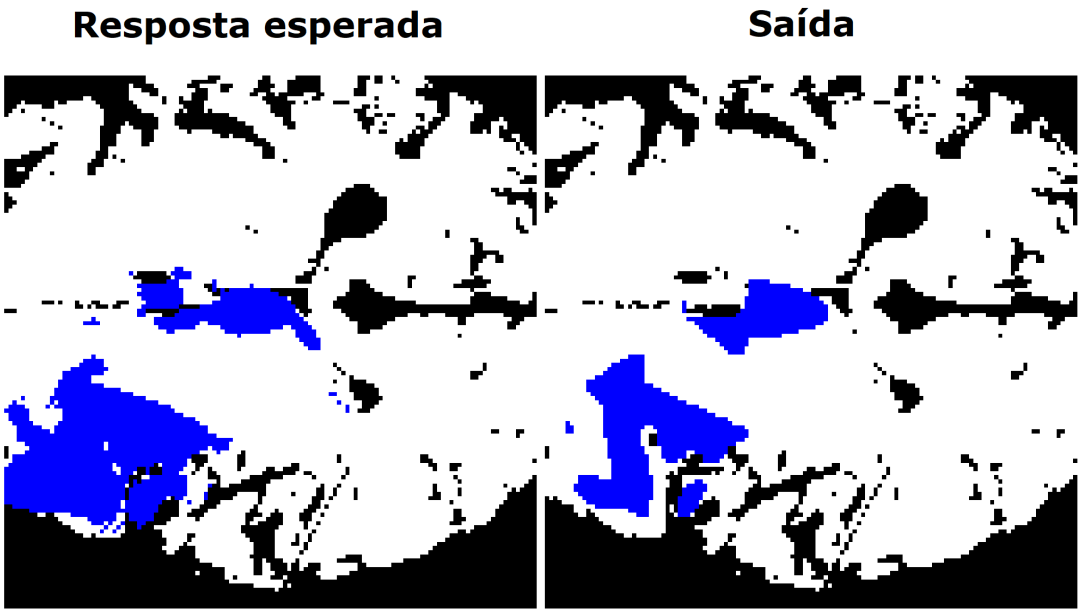
elevada quanto o demonstrado nas imagens. Os experimentos introduziram propositalmente um nível maior de intensidade para estressar o máximo possível a arquitetura e verificar sua resposta. Além disso, é importante ressaltar que essa simulação foi feita com apenas um tipo de degradação, enquanto que na prática o comum é haver algumas delas simultaneamente. Quando tipos diferentes de degradação se juntam, a tarefa se torna mais complexa, o PSNR diminui e o modelo tende a cometer mais falhas.



(a) Sem ruído

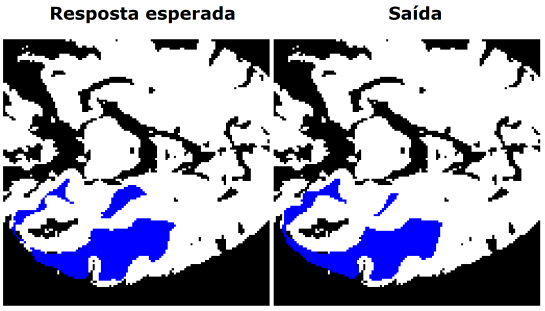


(b) Ruído gaussiano com desvio padrão de 0,325

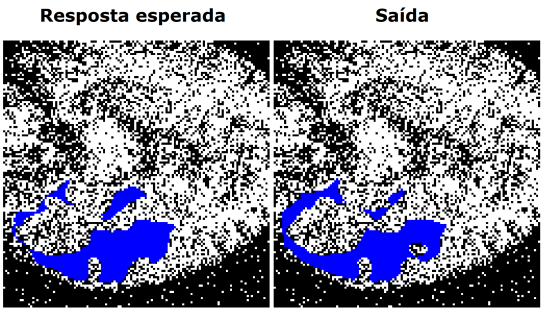


(c) Degradação por movimento de nível 5 na captura da imagem

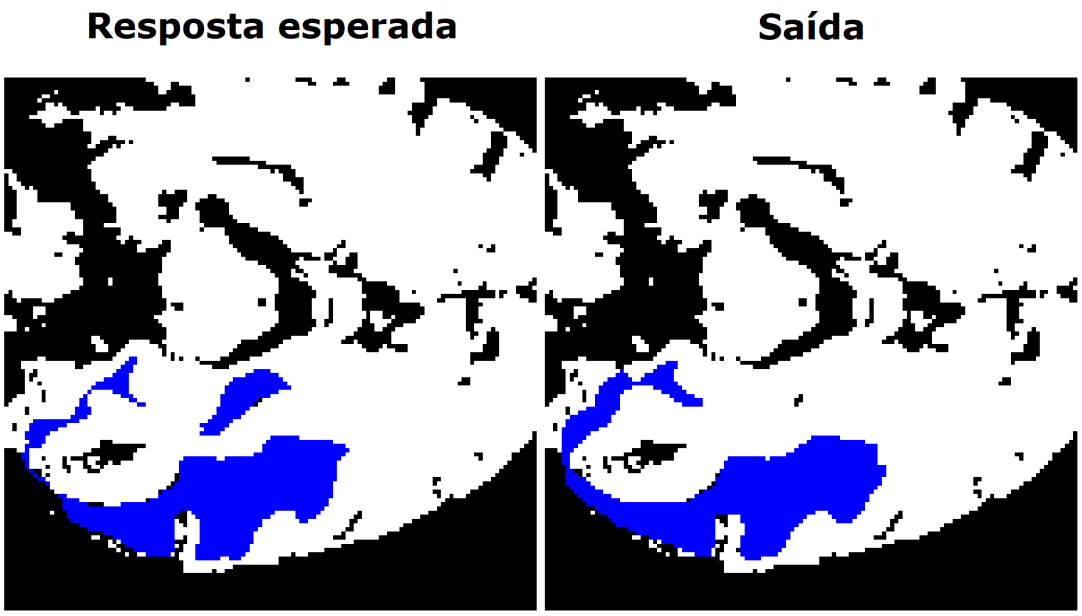
Figura 25 – Imagens das identificações e segmentações das áreas com tumores obtidas com a arquitetura DMFnet para: (a) imagems sem degradações, (b) imagem com ruído e (c) imagem com borrado de movimento.



(a) Sem ruído

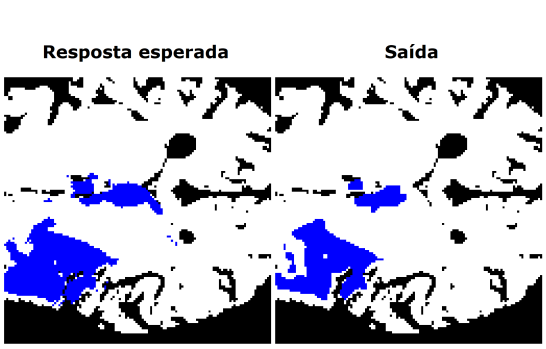


(b) Ruído gaussiano com desvio padrão de 0,325

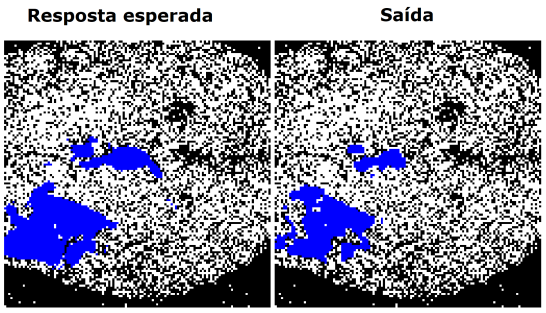


(c) Degradação por movimento de nível 5 na captura da imagem

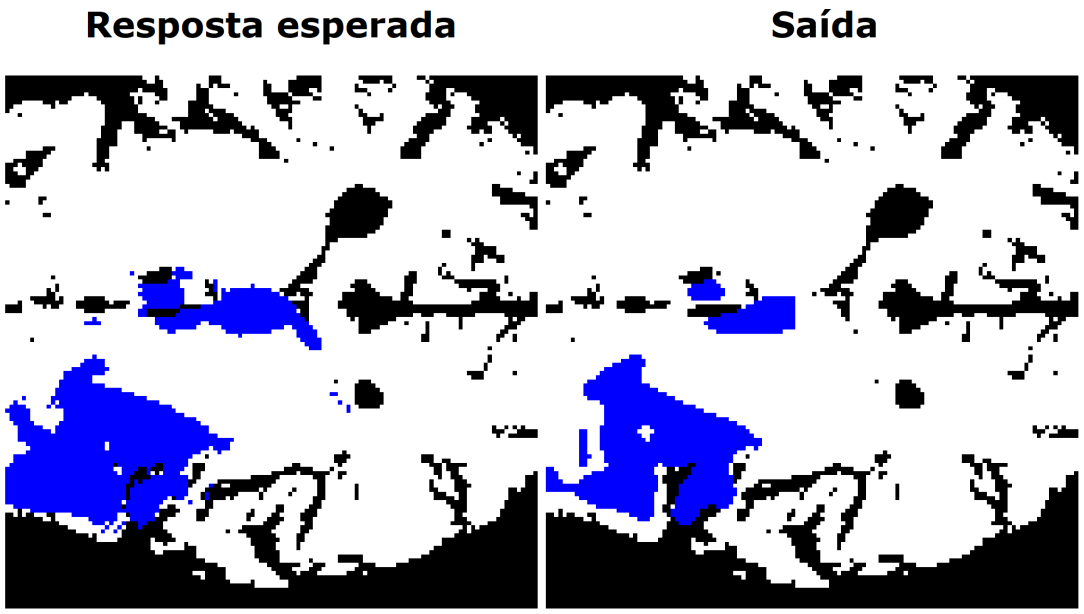
Figura 26 – Imagens das identificações e segmentações das áreas com tumores obtidas com a arquitetura V-net para: (a) imagems sem degradações, (b) imagem com ruído e (c) imagem com borrado de movimento.



(a) Sem ruído



(b) Ruído gaussiano com desvio padrão de 3,25



(c) Degradação por movimento de nível 5 na captura da imagem

Figura 27 – Imagens das identificações e segmentações das áreas com tumores obtidas com a arquitetura Residual U-net para: (a) imagems sem degradações, (b) imagem com ruído e (c) imagem com borrado de movimento.

5 Conclusão

5.1 Resultados gerais

Um dos objetivos do projeto é verificar qual a melhor arquitetura sem degradações, e a robustez dessas arquiteturas a degradações. Primeiramente, é importante conhecer o impacto dos hiper-parâmetros e otimizadores no desempenho. Nesse sentido, observou-se uma melhora grande ao usar o RMSProp ou o Adam, em vez do SGD. Os outros hiper-parâmetros não foram variados nesse trabalho.

Quando comparamos os resultados obtidos antes da introdução de degradações com as métricas publicadas nos artigos, observamos que a métrica Dice é um pouco menor e a distância de Hausdorff, um pouco maior que o visto na literatura. Assim, ambas as métricas indicam que o desempenho do modelo está menor que o esperado [Ballestar e Vilaplana 2020] [Chen et al. 2019]. Há algumas possíveis causas para isso:

- A inicialização dos pesos, que nesse trabalho foi aleatória;
- O tamanho do lote (*batch size*) menor que o utilizado na referência por limitações de RAM no trabalho;
- As técnicas de *data augmentation*, que tiveram parâmetros um pouco diferentes das técnicas usadas nos trabalhos originais;
- E o algoritmo de otimização, que não foi bem explicitado nos artigos.

Além disso, percebe-se, pelos gráficos, que a média de cada arquitetura apresentou valores próximos. Em relação a variação das métricas com relação ao PSNR, notou-se que a que menos varia é a V-net e a que mais varia, a Residual U-net. As classes também tiveram uma sensibilidade diferente, visto que a classe 1 (NET) possui uma sensibilidade maior e um desempenho pior que a classe 4 (ET).

Ressalta-se a contribuição desse trabalho para o meio científico: Os artigos estudados não levaram em consideração a possibilidade de imagens serem degradadas, o que não é realístico. Assim, esse trabalho vai além ao considerar essa possibilidade e demonstra como a degradação tem um efeito preocupante na segmentação semântica. Por exemplo, nas simulações da degradação por movimento, o tipo de degradação que mais diminui o desempenho da rede, quando aumentamos o número de movimentos de 1 para 10 do paciente, a Residual U-net apresenta uma queda de 16% na média do coeficiente Dice e um aumento de 11,1 da média da distância de Hausdorff. Na mesma simulação, para a V-net, a queda da média do coeficiente Dice foi de 9% e o aumento da média da distância

de Hausdorff foi de 3,8. Além disso, houve uma queda de 10 no PSNR para ambas as arquiteturas.

5.2 Perspectivas

Ressalta-se que esse projeto mostra um estudo das degradações frente a diversas arquiteturas. Contudo, é possível fazer outros experimentos para ter uma entendimento ainda maior do efeito da degradação nas arquiteturas:

- Treinar as arquiteturas com degradações e verificar se isso melhora ou não o desempenho no conjunto de testes com degradação;
- Verificar técnicas de retirada de ruídos, como os filtros, e como as arquiteturas se comportam diante desse novo cenário;
- Usar mais de uma degradação ao mesmo tempo e verificar o quanto isso prejudica o desempenho.

Referências

ACHARYA, U. R. et al. A deep convolutional neural network model to classify heartbeats. *Computers in Biology and Medicine*, v. 89, p. 389–396, 2017. ISSN 0010-4825. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0010482517302810>. Citado na página 16.

ALI, H. M. A new method to remove salt amp; pepper noise in magnetic resonance images. In: *2016 11th International Conference on Computer Engineering Systems (ICCES)*. [S.l.: s.n.], 2016. p. 155–160. Citado na página 17.

BAKAS, S. et al. *Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge*. 2019. Citado na página 21.

BALLESTAR, L. M.; VILAPLANA, V. *MRI brain tumor segmentation and uncertainty estimation using 3D-UNet architectures*. 2020. Citado 7 vezes nas páginas 11, 24, 25, 27, 29, 32 e 39.

CASTRO, E.; CARDOSO, J. S.; PEREIRA, J. C. Elastic deformations for data augmentation in breast cancer mass detection. In: *2018 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*. [S.l.: s.n.], 2018. p. 230–234. Citado na página 13.

CHEN, C. et al. *3D Dilated Multi-Fiber Network for Real-time Brain Tumor Segmentation in MRI*. 2019. Citado 7 vezes nas páginas 12, 13, 25, 26, 29, 32 e 39.

CHOI, D. et al. *On Empirical Comparisons of Optimizers for Deep Learning*. 2020. Citado na página 9.

CIRILLO, M. D.; ABRAMIAN, D.; EKLUND, A. *What is the best data augmentation for 3D brain tumor segmentation?* 2021. Citado na página 13.

DONG, H. et al. Automatic brain tumor detection and segmentation using u-net based fully convolutional networks. In: HERNÁNDEZ, M. V.; GONZÁLEZ-CASTRO, V. (Ed.). *Medical Image Understanding and Analysis*. Cham: Springer International Publishing, 2017. p. 506–517. ISBN 978-3-319-60964-5. Citado na página 2.

FIDON, L. et al. Generalised wasserstein dice score for imbalanced multi-class segmentation using holistic convolutional networks. *Lecture Notes in Computer Science*, Springer International Publishing, p. 64–76, 2018. ISSN 1611-3349. Disponível em: <http://dx.doi.org/10.1007/978-3-319-75238-9_6>. Citado na página 8.

FIDON, L.; OURSELIN, S.; VERCAUTEREN, T. Generalized wasserstein dice score, distributionally robust deep learning, and ranger for brain tumor segmentation: Brats 2020 challenge. *Lecture Notes in Computer Science*, Springer International Publishing, p. 200–214, 2021. ISSN 1611-3349. Disponível em: <http://dx.doi.org/10.1007/978-3-030-72087-2_18>. Citado na página 8.

GEISSLER, A. et al. Differential functional benefits of ultra highfield mr systems within the language network. *NeuroImage*, v. 103, 09 2014. Citado na página 17.

GODENSCHWEGER, F. et al. Motion correction in mri of the brain. *Phys Med Biol*, 2016. Citado na página 18.

GONZALEZ, R. C.; WOODS, R. E. *Digital image processing*. Upper Saddle River, N.J.: Prentice Hall, 2008. ISBN 9780131687288 013168728X 9780135052679 013505267X. Disponível em: <http://www.amazon.com/Digital-Image-Processing-3rd-Edition/dp/013168728X>. Citado na página 16.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado 2 vezes nas páginas 5 e 6.

GRAVES, A. *Generating Sequences With Recurrent Neural Networks*. 2014. Citado 2 vezes nas páginas 9 e 12.

GRAVES, M.; DONALD, M. Body mri artifacts in clinical practice: a physicist's and radiologist's perspective. *J Magn Reson Imaging*, 2013. PMID: 23960007. Citado na página 17.

GURESEN, E.; KAYAKUTLU, G. Definition of artificial neural networks with comparison to other networks. *Procedia Computer Science*, v. 3, p. 426–433, 2011. ISSN 1877-0509. World Conference on Information Technology. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050910004461>. Citado na página 6.

HE, K. et al. *Deep Residual Learning for Image Recognition*. 2015. Citado na página 25.

HENRY, T. et al. *Brain tumor segmentation with self-ensembled, deeply-supervised 3D U-net neural networks: a BraTS 2020 challenge solution*. 2020. Citado na página 24.

JORRITSMA, W.; CNOSSEN, F.; van Ooijen, P. Improving the radiologist–cad interaction: designing for appropriate trust. *Clinical Radiology*, v. 70, n. 2, p. 115–122, 2015. ISSN 0009-9260. Disponível em: <https://www.sciencedirect.com/science/article/pii/S000992601400453X>. Citado na página 2.

KAUR, P.; SINGH, J. A study on the effect of gaussian noise on psnr value for digital images. *International Journal of Computer and Electrical Engineering*, p. 319–321, 2011. Citado na página 18.

KERVADEC, H. et al. Boundary loss for highly unbalanced segmentation. In: CARDOSO, M. J. et al. (Ed.). *Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning*. PMLR, 2019. (Proceedings of Machine Learning Research, v. 102), p. 285–296. Disponível em: <https://proceedings.mlr.press/v102/kervadec19a.html>. Citado na página 23.

KHAN, A. et al. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, Springer Science and Business Media LLC, v. 53, n. 8, p. 5455–5516, Apr 2020. ISSN 1573-7462. Disponível em: <http://dx.doi.org/10.1007/s10462-020-09825-6>. Citado na página 16.

KINGMA, D. P.; BA, J. *Adam: A Method for Stochastic Optimization*. 2017. Citado 3 vezes nas páginas 9, 11 e 12.

LI, F.-F.; JOHNSON, J.; YEUNG, S. Lecture 11: Detection and segmentation. 2017. Acesso em: 01/11/2021. Disponível em: <http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf>. Citado 3 vezes nas páginas 11, 3 e 4.

LIASHCHYNSKYI, P.; LIASHCHYNSKYI, P. *Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS*. 2019. Citado na página 5.

LIN, H. Identification of spinal deformity classification with total curvature analysis and artificial neural network. *IEEE Transactions on Biomedical Engineering*, v. 55, n. 1, p. 376–382, 2008. Citado na página 9.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015. Citado na página 16.

MILLETARI, F.; NAVAB, N.; AHMADI, S.-A. *V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation*. 2016. Citado 3 vezes nas páginas 13, 11 e 27.

MYRONENKO, A. *3D MRI brain tumor segmentation using autoencoder regularization*. 2018. Citado 2 vezes nas páginas 11 e 24.

NACEUR, M. B. et al. A new online class-weighting approach with deep neural networks for image segmentation of highly unbalanced glioblastoma tumors. In: ROJAS, I.; JOYA, G.; CATALA, A. (Ed.). *Advances in Computational Intelligence*. Cham: Springer International Publishing, 2019. p. 555–567. ISBN 978-3-030-20518-8. Citado na página 23.

PANCHAPAGESAN, S. et al. Multi-task learning and weighted cross-entropy for dnn-based keyword spotting. In: *INTERSPEECH*. [S.l.: s.n.], 2016. Citado na página 8.

PÉREZ-GARCÍA, F.; SPARKS, R.; OURSELIN, S. Torchio: a python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning. *Computer Methods and Programs in Biomedicine*, p. 106236, 2021. ISSN 0169-2607. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0169260721003102>. Citado na página 28.

RAICHLE, M. E. Two views of brain function. *Trends in Cognitive Sciences*, v. 14, n. 4, p. 180–190, 2010. ISSN 1364-6613. Disponível em: <https://www.sciencedirect.com/science/article/pii/S136466131000029X>. Citado na página 1.

RAWAT, W.; WANG, Z. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*, v. 29, n. 9, p. 2352–2449, 09 2017. ISSN 0899-7667. Disponível em: <https://doi.org/10.1162/neco_a_00990>. Citado 2 vezes nas páginas 14 e 16.

RIBERA, J. et al. *Locating Objects Without Bounding Boxes*. 2019. Citado na página 20.

ROY, S. et al. *A Review on Automated Brain Tumor Detection and Segmentation from MRI of Brain*. 2013. Citado na página 1.

SMITH, L. N. A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay. *CoRR*, abs/1803.09820, 2018. Disponível em: <http://arxiv.org/abs/1803.09820>. Citado na página 9.

TANG, Y. et al. Svms modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, v. 39, n. 1, p. 281–288, 2009. Citado na página 23.

TIU, E. Metrics to evaluate your semantic segmentation model. 2019. Acesso em: 20/10/2021. Disponível em: <https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2>. Citado na página 19.

TYAGI, V. A review on image classification techniques to classify neurological disorders of brain mri. In: *2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*. [S.l.: s.n.], 2019. v. 1, p. 1–4. Citado 2 vezes nas páginas 3 e 6.

WILD, C.; WEIDERPASS, E.; STEWART, B. World cancer report. 2020. Citado 2 vezes nas páginas 11 e 2.

WU, S. et al. *IoU-balanced Loss Functions for Single-stage Object Detection*. 2020. Citado na página 8.

Yang, G. et al. Dagan: Deep de-aliasing generative adversarial networks for fast compressed sensing mri reconstruction. *IEEE Transactions on Medical Imaging*, v. 37, n. 6, p. 1310–1321, 2018. Citado na página 2.

YU, C. et al. *BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation*. 2018. Citado na página 11.

ZHANG, S. et al. Recursive convolution neural network for polsar image classification. In: *2020 35th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. [S.l.: s.n.], 2020. p. 482–485. Citado na página 16.

ZHAO, L.; JIA, K. Deep feature learning with discrimination mechanism for brain tumor segmentation and diagnosis. In: *2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*. [S.l.: s.n.], 2015. p. 306–309. Citado na página 2.

ZULKIFLI, H. Understanding learning rates and how it improves performance in deep learning. 2018. Acesso em: 10/10/2021. Disponível em: <https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10>. Citado na página 10.

ÇIçEK Özgün et al. *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation*. 2016. Citado 2 vezes nas páginas 12 e 24.