



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CAMPUS FLORIANÓPOLIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Felipe Sassi Del Sant

**Redes neurais convolucionais para detecção do *status* do gene MGMT em pacientes com glioblastomas a partir de imagens de ressonância magnética**

Florianópolis  
2022

Felipe Sassi Del Sant

**Redes neurais convolucionais para detecção do *status* do gene MGMT em pacientes com glioblastomas a partir de imagens de ressonância magnética**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina para a obtenção do título de mestre em Engenharia elétrica.

Orientador: Prof. Danilo Silva, Ph.D.

Florianópolis

2022

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Del Sant, Felipe Sassi

Redes neurais convolucionais para detecção do status do gene MGMT em pacientes com glioblastomas a partir de imagens de ressonância magnética / Felipe Sassi Del Sant ; orientador, Danilo Silva, 2022.

88 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia Elétrica, Florianópolis, 2022.

Inclui referências.

1. Engenharia Elétrica. 2. Classificação do status do gene MGMT. 3. Aprendizado de máquina. 4. Aprendizado profundo. 5. Redes neurais convolucionais. I. Silva, Danilo. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

Felipe Sassi Del Sant

**Redes neurais convolucionais para detecção do *status* do gene MGMT em pacientes com glioblastomas a partir de imagens de ressonância magnética**

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Márcio Holsbach Costa, Ph.D.  
Universidade Federal de Santa Catarina

Carlos Eduardo Martins Relvas, Dr.  
Datarisk

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em Engenharia elétrica.

---

Telles Brunelli Lazzarin, Dr.  
Coordenador do Programa de Pós-Graduação  
em Engenharia Elétrica

---

Prof. Danilo Silva, Ph.D.  
Orientador

Florianópolis, 2022.



Dedicado a todos que me auxiliaram durante essa jornada.

---

## Agradecimentos

---

Agradeço a todos os que contribuíram direta e indiretamente para o sucesso desse trabalho.

*“Estude muito o que mais lhe interessa  
da maneira mais indisciplinada,  
irreverente e original possível.”  
FEYNMAN, 1965*

## RESUMO

A automatização de diferentes processos médicos via aplicação de aprendizado de máquina vem ganhando destaque ao longo do tempo, devido a seus resultados equiparáveis aos de especialistas em diversas áreas. Nesse trabalho, técnicas de aprendizado de máquina são aplicadas na tarefa de classificação do *status* do gene MGMT em pacientes com glioblastomas, uma categoria de tumor cerebral com elevado nível de agressividade, com um tempo médio de sobrevivência de cerca de 14,6 meses. Métodos cirúrgicos, complementados por tratamentos quimioterápicos, visam aumentar o tempo de sobrevivência dos pacientes e melhorar seu bem-estar. O sucesso desses tratamentos está ligado a determinados marcadores genéticos, como o nível de metilação (silenciamento das atividades) do gene MGMT. Métodos invasivos são utilizados para identificação do *status* do gene, os quais acarretam riscos para o paciente, limitando sua aplicabilidade. Devido a isso, técnicas não-invasivas, baseadas na classificação de imagens de ressonância magnética por meio de aprendizado de máquina, vêm sendo desenvolvidas. Esse trabalho explora quatro técnicas para classificação do *status* do gene MGMT em pacientes com glioblastomas, visando a comparação dos resultados de cada método em uma base de dados pública, comum e de elevada qualidade. Especificamente, são investigadas abordagens baseadas em extração de *features* radiômicas e baseadas em redes neurais convolucionais, tanto para classificação 2D/3D quanto para segmentação semântica. Nenhuma das abordagens testadas demonstrou significativa capacidade de classificação automática do *status* do gene MGMT, todas tendo, em média, uma ROC-AUC (%) de 60% ( $\pm 2,78\%$ ) no conjunto de teste utilizado para comparação dos resultados.

**Palavras-chave:** Classificação do *status* do gene MGMT. Gene MGMT. Aprendizado de máquina. Aprendizado profundo. Redes neurais convolucionais.

## ABSTRACT

The automation of different medical processes via the application of machine learning has been gaining prominence over time, due to its results comparable to those of experts in various fields. In this work, machine learning techniques are applied to the task of classifying the status of the MGMT gene in patients with glioblastomas, a category of brain tumor with a high level of aggressiveness, with an average survival time of about 14.6 months. Surgical methods, complemented by chemotherapeutic treatments, aim to increase the survival time of patients and improve their well-being. The success of these treatments is linked to certain genetic markers, such as the level of methylation (silencing activities) of the MGMT gene. Invasive methods are used to identify the gene status, which carry risks for the patient that limit their applicability. Because of this, non-invasive techniques, based on classification of magnetic resonance images using machine learning, have been developed. This paper explores four techniques for classifying MGMT gene status in patients with glioblastomas, aiming to compare the results of each method on a common, high quality, public database. Specifically, radiomic feature extraction-based and convolutional neural network-based approaches for both 2D/3D classification and semantic segmentation are investigated. None of the approaches tested demonstrated much ability to automatically classify the status of the MGMT gene, all having, on average, an ROC-AUC (%) of 60% ( $\pm 2,78\%$ ) on the test set used to compare results.

**Keywords:** Classification of MGMT gene *status*. MGMT gene. Machine learning. Deep learning. Convolutional neural networks.

## LISTA DE FIGURAS

Figura 1 – Exemplos de segmentação de tumor em diferentes cortes de uma RMI. . . . .	16
Figura 2 – Exemplificação da construção de uma árvore de decisão a partir de um conjunto de dados simples. . . . .	23
Figura 3 – Resultado da aplicação de uma máscara a uma imagem de entrada. . . . .	24
Figura 4 – A clusterização visa encontrar grupos semelhantes num espaço de <i>features</i> de entrada. . . . .	25
Figura 5 – Exemplo de validação cruzada de 5 <i>folds</i> . . . . .	26
Figura 6 – Representação dos valores VP, VN, FP, FN em uma matriz de confusão. . . . .	27
Figura 7 – Curva ROC de exemplo e reta identificando o desempenho aleatório. . . . .	28
Figura 8 – Exemplo de rede neural densa com 2 entradas e 1 saída. . . . .	29
Figura 9 – Exemplo da operação de convolução entre uma imagem de entrada e um <i>kernel</i> . . . . .	31
Figura 10 – Exemplificação do funcionamento da camada de <i>max pooling</i> . . . . .	31
Figura 11 – Bloco construtivo da arquitetura ResNet. . . . .	33
Figura 12 – Arquitetura <i>encoder-decoder</i> utilizada para segmentação semântica. . . . .	34
Figura 13 – Exemplo de convolução transposta. . . . .	34
Figura 14 – Exemplo da aplicação de <i>dropout</i> em um modelo de <i>deep learning</i> . . . . .	36
Figura 15 – Exemplo da aplicação do registro a uma RMI. . . . .	38
Figura 16 – Arquitetura NN-UNet utilizada para segmentação de tumores cerebrais. . . . .	40
Figura 17 – Ilustração da abordagem baseada em segmentação semântica. . . . .	47
Figura 18 – Exemplo de sequências que compõe o conjunto de dados BRATS 2021. . . . .	49
Figura 19 – <i>Pipeline</i> utilizada na abordagem baseada em extração de <i>features</i> . . . . .	52
Figura 20 – Exemplificação do esquema de validação utilizado nessa abordagem. . . . .	53
Figura 21 – <i>Pipeline</i> utilizada na abordagem <i>end-to-end</i> 3D. . . . .	54
Figura 22 – Organização das fatias em uma RMI. . . . .	54
Figura 23 – <i>Pipeline</i> utilizada durante a abordagem baseada em fatias. . . . .	55
Figura 24 – Diagrama representando a abordagem de classificação via segmentação semântica. . . . .	56
Figura 25 – Exemplificação da extração de <i>patches</i> a partir de uma RMI. . . . .	57
Figura 26 – Resultados da métrica ROC-AUC (%) ao longo das simulações para o modelo de regressão logística. . . . .	61
Figura 27 – Resultados da métrica ROC-AUC ao longo das 100 simulações para o modelo de <i>gradient boosting</i> . . . . .	62
Figura 28 – Resultados da métrica ROC-AUC ao longo das simulações para o modelo de <i>random forests</i> . . . . .	62
Figura 29 – Resultados da métrica ROC-AUC (%) no conjunto de testes ao longo das simulações com diferentes modelos. . . . .	63

Figura 30 – Comparação da curva ROC para cada um dos modelos. . . . .	64
Figura 31 – Resultados da aplicação da técnica de <i>data augmentation</i> aplicada durante o treinamento. . . . .	66
Figura 32 – Custo ao longo das épocas nos conjuntos de treinamento e validação para a abordagem <i>end-to-end</i> 3D. . . . .	67
Figura 33 – Métrica ROC-AUC (%) ao longo das épocas nos conjuntos de treinamento e validação para a abordagem <i>end-to-end</i> 3D. . . . .	67
Figura 34 – Curvas ROC para os conjuntos de treino, validação e teste. . . . .	68
Figura 35 – Resultados de uma rotação aleatória aplicada na imagem original. . . . .	69
Figura 36 – Resultados da adição de um ruído gaussiano na imagem original. . . . .	70
Figura 37 – Evolução da função custo ao longo das épocas nos conjuntos de treinamento e validação para a abordagem baseada em fatias. . . . .	71
Figura 38 – Métrica ROC-AUC (%) ao longo das épocas nos conjuntos de treinamento e validação para a abordagem baseada em fatias. . . . .	72
Figura 39 – Curvas ROC o experimento de classificação 2D baseada em fatias. . . . .	73
Figura 40 – Resultado das transformações de <i>data augmentation</i> utilizadas na abordagem baseada em segmentação semântica. . . . .	75
Figura 41 – Evolução da função custo ao longo das épocas nos conjuntos de treinamento e validação para a abordagem baseada em segmentação semântica sem pré-treinamento. . . . .	76
Figura 42 – Evolução da métrica ROC-AUC (%) ao longo das épocas no conjunto de treinamento para a abordagem baseada em segmentação semântica sem pré-treinamento. . . . .	77
Figura 43 – Evolução da métrica ROC-AUC (%) ao longo das épocas no conjunto de validação para a abordagem baseada em segmentação semântica sem pré-treinamento. . . . .	77
Figura 44 – Curvas ROC para a abordagem baseada em segmentação semântica. . . . .	78
Figura 45 – Evolução da função custo ao longo das épocas nos conjuntos de treinamento e validação para a abordagem baseada em segmentação semântica com pré-treinamento. . . . .	79
Figura 46 – Evolução da métrica ROC-AUC ao longo das épocas no conjunto de treinamento para a abordagem baseada em segmentação semântica com pré-treinamento. . . . .	80
Figura 47 – Evolução da métrica ROC-AUC ao longo das épocas no conjunto de validação para a abordagem baseada em segmentação semântica com pré-treinamento. . . . .	80
Figura 48 – Curvas ROC para a abordagem baseada em segmentação semântica. . . . .	81
Figura 49 – Curvas ROC para os conjuntos de treino, validação e teste. . . . .	84

## LISTA DE TABELAS

Tabela 1 – Resultados em termos de ROC-AUC (%) obtidos na abordagem baseada em extração de <i>features</i> . . . . .	64
Tabela 2 – Hiperparâmetros obtidos na abordagem <i>end-to-end</i> . . . . .	66
Tabela 3 – Resultados em termos de métrica ROC-AUC (%) obtidos na abordagem <i>end-to-end</i> 3D . . . . .	68
Tabela 4 – Hiperparâmetros obtidos na abordagem baseada em classificação de fatias .	70
Tabela 5 – Resultados em termos de ROC-AUC (%) obtidos na abordagem baseadas em fatias . . . . .	72
Tabela 6 – Hiperparâmetros obtidos na abordagem baseada em segmentação semântica	75
Tabela 7 – Resultados obtidos em termos de ROC-AUC (%) na abordagem baseadas em segmentação semântica . . . . .	78
Tabela 8 – Resultados obtidos em termos de ROC-AUC (%) na abordagem baseadas em segmentação semântica com pré-treinamento . . . . .	81
Tabela 9 – Resultados em termos de ROC-AUC (%) obtidos na abordagem baseadas em segmentação semântica no conjunto de dados TCIA . . . . .	82
Tabela 10 – Comparação em termos de métrica ROC-AUC (%) dos resultados obtidos ao longo dos experimentos . . . . .	83



---

## Lista de abreviaturas e siglas

---

I.A.	Inteligência artificial
MICCAI	<i>Medical Image Computing and Computer Assisted Intervention Society</i>
RMI	Imagem de ressonância magnética
ROC	<i>Receiver operating characteristic curve</i>
ROI	<i>Region of interest</i>
RSNA	<i>Radiological Society of North America</i>
TCIA	<i>The Image Cancer Archive</i>
TMZ	Temozolomida

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	OBJETIVOS	18
<b>1.1.1</b>	<b>Objetivo geral</b>	<b>18</b>
<b>1.1.2</b>	<b>Objetivos específicos</b>	<b>18</b>
1.2	CONTRIBUIÇÕES	19
1.3	ORGANIZAÇÃO	19
<b>2</b>	<b>CONCEITOS BÁSICOS</b>	<b>20</b>
2.1	APRENDIZADO DE MÁQUINA	20
<b>2.1.1</b>	<b>Aprendizado supervisionado</b>	<b>21</b>
<b>2.1.2</b>	<b>Aprendizado não-supervisionado</b>	<b>24</b>
2.2	AVALIAÇÃO DE MODELOS DE CLASSIFICAÇÃO BINÁRIA	25
2.3	APRENDIZADO PROFUNDO	29
<b>2.3.1</b>	<b>Redes neurais densas</b>	<b>29</b>
<b>2.3.2</b>	<b>Redes neurais convolucionais</b>	<b>30</b>
<b>2.3.3</b>	<b>Treinamento de redes neurais</b>	<b>34</b>
2.4	RESSONÂNCIAS MAGNÉTICAS	36
2.5	PRÉ-PROCESSAMENTO DE RESSONÂNCIAS MAGNÉTICAS	37
<b>2.5.1</b>	<b>Co-registro de volumes</b>	<b>38</b>
<b>2.5.2</b>	<b>Remoção de crânio</b>	<b>38</b>
<b>2.5.3</b>	<b>Segmentação de tumores cerebrais</b>	<b>39</b>
<b>2.5.4</b>	<b>Extração de <i>features</i> de tumores cerebrais</b>	<b>40</b>
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>42</b>
3.1	CLASSIFICAÇÃO ATRAVÉS DE EXTRAÇÃO DE <i>FEATURES</i>	42
3.2	CLASSIFICAÇÃO <i>END-TO-END</i> COM <i>DEEP LEARNING</i>	43
3.3	<i>DEEP LEARNING</i> PARA CLASSIFICAÇÃO DE FATIAS	44
3.4	ABORDAGEM VIA SEGMENTAÇÃO SEMÂNTICA	45
<b>4</b>	<b>METODOLOGIA</b>	<b>48</b>
4.1	CONJUNTOS DE DADOS	48
<b>4.1.1</b>	<b>Divisão do conjunto de dados</b>	<b>50</b>
4.2	ABORDAGENS MAPEADAS	51
<b>4.2.1</b>	<b>Abordagem baseada em extração de <i>features</i></b>	<b>52</b>
<b>4.2.2</b>	<b>Abordagem 3D (<i>end-to-end</i>)</b>	<b>53</b>
<b>4.2.3</b>	<b>Abordagem 2D (baseada em fatias)</b>	<b>54</b>
<b>4.2.4</b>	<b>Abordagem via segmentação semântica</b>	<b>56</b>
<b>5</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>59</b>
5.1	CLASSIFICAÇÃO A PARTIR DE <i>FEATURES</i> RADIÔMICAS	60
5.2	CLASSIFICAÇÃO 3D <i>END-TO-END</i>	64

5.3	CLASSIFICAÇÃO 2D BASEADA EM FATIAS . . . . .	68
5.4	CLASSIFICAÇÃO A PARTIR DE SEGMENTAÇÃO SEMÂNTICA . . . . .	73
<b>5.4.1</b>	<b>Sem pré-treinamento . . . . .</b>	<b>73</b>
<b>5.4.2</b>	<b>Com pré-treinamento . . . . .</b>	<b>78</b>
<b>5.4.3</b>	<b>Resultados no conjunto de dados TCIA . . . . .</b>	<b>82</b>
5.5	COMPARAÇÃO DE ABORDAGENS E DISCUSSÃO . . . . .	83
	<b>6 CONCLUSÃO . . . . .</b>	<b>86</b>
6.1	TRABALHOS FUTUROS . . . . .	87
	<b>REFERÊNCIAS . . . . .</b>	<b>88</b>

# CAPÍTULO 1

---

## Introdução

---

Aprendizado de máquina é um campo da área de inteligência artificial (I.A.). Esse campo explora a modelagem/previsão de fenômenos a partir do ajuste de modelos matemáticos em conjunto de dados relacionados ao problema em análise. Atualmente, aprendizado de máquina encobre outra área conhecida como aprendizado profundo, a qual constrói modelos baseados em redes neurais artificiais para resolução de diferentes categorias de problemas.

Para auxiliar no diagnóstico médico, soluções envolvendo aprendizado de máquina vêm sendo desenvolvidas e vêm ganhando cada vez mais destaque na sociedade. Essas soluções muitas vezes são desenvolvidas com base em aprendizado profundo devido à natureza não estruturada dos dados (imagens, texto, etc.). Esses projetos utilizam o potencial desses modelos para realização de diferentes tarefas como segmentação de órgãos (GOYAL *et al.*, 2022), localização de nódulos (ISENSEE *et al.*, 2020) e classificação de patologias de maneira automatizada, servindo como ferramenta de auxílio ao diagnóstico.

Uma das possíveis aplicações de I.A. é no diagnóstico automatizado de gliomas, tumores cerebrais que afetam células da glia (cuja função é proteger e nutrir neurônios). Esse tipo de neoplasia (massa de tecido anormal) é comumente encontrado sob a forma de glioblastoma multiforme, a qual também é a categoria mais agressiva de tumor cerebral (GLIOBLASTOMA, 2018). Cerca de 45% dos tumores cerebrais que acometem o sistema nervoso central são glioblastomas multiformes, com um tempo médio de sobrevivência de 14,6 meses (JOHNSON; O'NEILL, 2012).

Os dados utilizados nas soluções automatizadas para diagnóstico de gliomas são geralmente oriundos de imagens de ressonâncias magnéticas (RMI) do cérebro. A RMI é uma das principais ferramentas para diagnóstico de tumores envolvendo o cérebro devido à possibilidade de distinção entre tumores e tecido cerebral saudável. Além disso, ela permite obter detalhes completos sobre a estrutura do tumor, bem como características descritivas úteis para

o diagnóstico.

Para localizar e identificar esses tumores uma das principais técnicas é a segmentação manual dos resultados da RMI. Essa técnica é custosa em termos de tempo e altamente dependente da experiência do médico, dificultando a conferência de resultados devido a ruídos presentes na segmentação de cada profissional. Esses problemas, inerentes a técnicas manuais, são a principal motivação para o desenvolvimento da segmentação automática de tumores cerebrais. Para o médico essa abordagem simplifica o processo de detecção da doença, bem como acelera o diagnóstico. Já para o paciente, a automatização do diagnóstico traz alguns benefícios, como possibilidade de diagnóstico precoce e preciso, o qual é útil para desenvolvimento de tratamentos mais eficazes e aumento da sobrevivida. A Figura 18 apresenta diferentes cortes de uma RMI e a segmentação de um glioma do tipo glioblastoma multiforme.

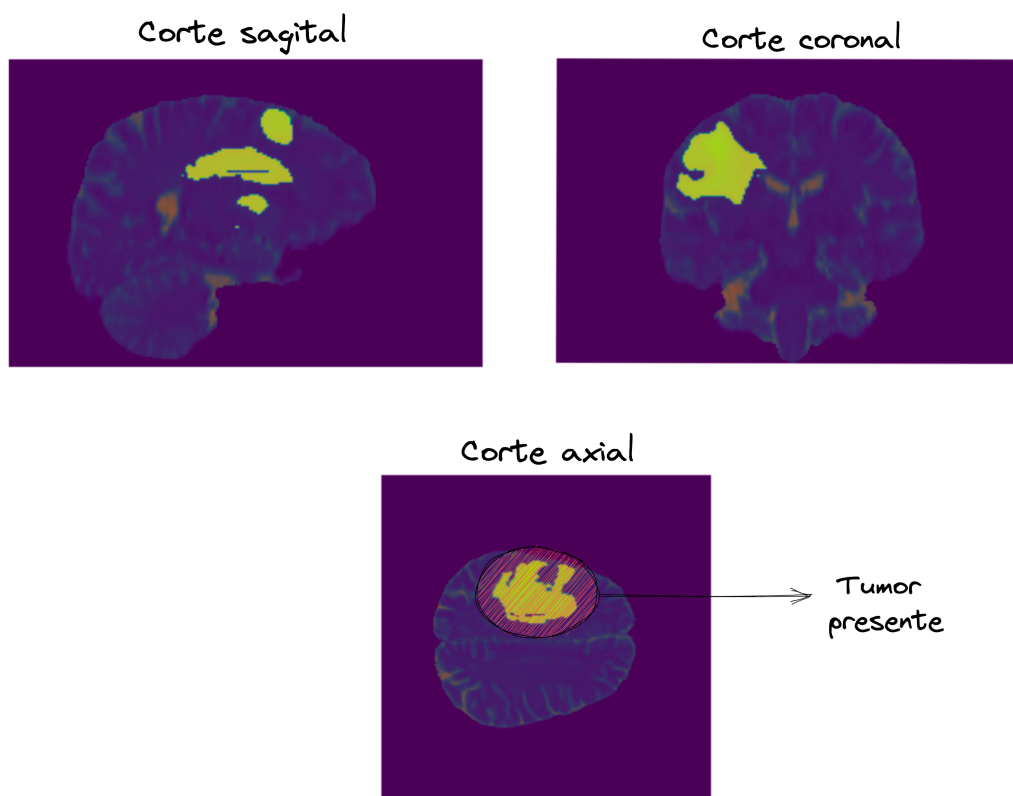


Figura 1 – Exemplos de segmentação de tumor em diferentes cortes de uma RMI.

A técnica de geração automática de segmentação é chamada de segmentação semântica a qual é desenvolvida em diferentes trabalhos para simplificação do processo de localização/segmentação de tumores cerebrais. Muitas delas atingindo resultados equiparáveis com comitês de profissionais da área (ISENSEE *et al.*, 2021), mostrando um cenário otimista para a aplicação dessas técnicas num futuro não muito distante.

A partir do conhecimento da doença e correta localização, seu tratamento é realizado tanto de forma cirúrgica quanto a partir de um tratamento quimioterápico. Atualmente, um dos principais tratamentos é a quimioterapia com agentes alquilantes, como a Temozolomida

(TMZ), o qual tem sua taxa de sucesso atrelada a determinados marcadores genéticos do paciente (TRINDADE *et al.*, 2012).

Um dos marcadores genéticos de interesse é o gene MGMT (*O6-metilguanina-DNA metiltransferase*) devido à sua relação com o prognóstico dos pacientes. Esse gene tem um efeito reparador nas células tumorais devido à resistência, proporcionada por ele, a agentes alquilantes. O estudo do gene MGMT está relacionado à identificação da metilação (silenciamento) desse gene, o qual provoca alterações que acabam removendo seu efeito restaurador em células tumorais (GEPP, 2013). A presença desse gene metilado está intimamente ligada ao sucesso de quimioterápicos alquilantes como TMZ, sendo comumente usados para complementar os resultados obtidos cirurgicamente (ALMEIDA *et al.*, 2019) e aumentar o tempo de sobrevivência dos pacientes (TRINDADE *et al.*, 2012).

Atualmente, a principal forma de detecção do *status* da metilação do gene MGMT é via biópsia do tecido cerebral, a qual acarreta riscos para a saúde do paciente, principalmente quando o tumor está localizado em área de difícil acesso. Devido a esses riscos, um significativo esforço está sendo aplicado ao desenvolvimento de técnicas não invasivas, baseadas em imagens e em características radiômicas para identificação do *status* do gene MGMT (YOGANANDA *et al.*, 2021) e consequente melhora no diagnóstico.

Para a área médica, a identificação automática pode trazer ganhos como redução de custos, aumento da velocidade de resposta e melhor personalização de tratamento (CHANG *et al.*, 2018), em relação aos métodos clássicos utilizados atualmente, como biópsias. Além disso, sua característica não-invasiva o torna especialmente importante em casos onde a cirurgia não é possível.

Devido à dificuldade e importância da detecção da metilação desse gene, uma competição envolvendo a classificação do *status* do gene a partir de imagens de ressonância magnética foi hospedada na plataforma *Kaggle*, com prêmios para os primeiros colocados. As entidades responsáveis pela manutenção da competição foram a *Radiological Society of North America* (RSNA) em conjunto com o *Medical Image Computing and Computer Assisted Intervention Society* (MICCAI).

Essa competição visou incentivar a comunidade mundial de ciência de dados a desenvolver estratégias para classificar com sucesso o *status* do gene MGMT. A métrica utilizada para comparação dos resultados foi a ROC-AUC (área abaixo da curva ROC). O melhor resultado obteve uma métrica ROC-AUC (%) de 62,17% (dentre mais de 1500 participantes), mostrando a dificuldade de se obter uma classificação assertiva somente a partir do conjunto de imagens do cérebro dos pacientes com tumor cerebral.

A dificuldade de se obter bons resultados nessa competição serviu como motivação para o trabalho (SAEED *et al.*, 2022) onde os autores exploram diferentes modelos de *deep learning* para gerar a classificação do *status* do gene MGMT, utilizando os dados da competição hospedada na plataforma *Kaggle*. Neste trabalho, tanto modelos 2D, quanto modelos 3D de classificação baseados em redes neurais convolucionais são explorados (as duas abordagens

mais utilizadas na competição) e seus resultados analisados.

No caso da classificação do *status* do gene MGMT, a abordagem baseada em modelos 3D obteve um resultado de 64% de métrica ROC-AUC (essa métrica foi medida em um conjunto de validação separado pelos autores do artigo, não sendo o mesmo conjunto dos finalistas da competição). Além da abordagem baseada em modelos de classificação 3D, os autores obtiveram resultados de ROC-AUC (%) próximos a 60% utilizando uma abordagem 2D.

A partir desse trabalho e dos resultados obtidos, os autores chegaram à conclusão de que não há indícios significativos de que a utilização somente de exames de ressonância magnética seja suficiente para realização de uma classificação acurada do *status* de metilação do gene MGMT.

No entanto, abordagens 2D e 3D não são as únicas possíveis de serem exploradas nesse problema. Alguns trabalhos (CHEN, T. *et al.*, 2015) utilizam técnicas de extração de *features* radiômicas (características numéricas relacionadas ao tumor) para posterior classificação do *status* do gene MGMT via técnicas de aprendizado de máquina, obtendo valores de métrica ROC-AUC (%) de 89,60%. Outros trabalhos adaptam modelos de segmentação semântica de tumores cerebrais (pré-treinados na classificação de outro marcador genético) para gerar a classificação do *status* do gene (YOGANANDA *et al.*, 2021), obtendo valores de ROC-AUC (%) de cerca de 90%. Ambas as abordagens não foram testadas em (SAEED *et al.*, 2022).

Essa discrepância de resultados entre diferentes trabalhos e entre diferentes abordagens, em conjunto com a relevância do problema tornam clara a motivação do trabalho: análise e verificação das estratégias realizadas em trabalhos anteriores (até mesmo as não tão exploradas) em uma base de dados comum, pública, e de elevada qualidade.

## 1.1 OBJETIVOS

Aqui, tanto os objetivos gerais quanto os objetivos específicos deste trabalho serão apresentados.

### 1.1.1 Objetivo geral

O objetivo geral desta dissertação é identificar a melhor forma, entre as apresentadas na literatura, para classificação do *status* do gene MGMT em pacientes com tumores cerebrais.

### 1.1.2 Objetivos específicos

- Identificar as metodologias utilizadas para classificação do *status* do gene MGMT;
- identificar as bases de dados disponíveis para essa tarefa;
- reproduzir as melhores abordagens presentes na literatura, incluindo a abordagem via extração de *features* e via segmentação semântica;

- validar diferentes abordagens em uma base de dados comum, obtendo uma comparação justa das técnicas;
- encontrar a melhor abordagem para classificação do *status* do gene MGMT.

## 1.2 CONTRIBUIÇÕES

A principal contribuição desta dissertação é na validação das diferentes metodologias propostas para classificação do *status* do gene MGMT, comparando as abordagens em um conjunto de dados padronizado, de alta qualidade e com a maior quantidade de exames (RMI) disponíveis.

## 1.3 ORGANIZAÇÃO

Esta dissertação está organizada da seguinte forma:

- no segundo capítulo são introduzidos conceitos básicos, necessários para o correto entendimento do trabalho. São revisados conceitos ligados a aprendizado de máquina, aprendizado profundo e avaliação de resultados. Conceitos ligados a imagens médicas e técnicas de pré-processamento também serão apresentados;
- o terceiro capítulo traz a revisão bibliográfica realizada. Nesse capítulo diferentes abordagens para a resolução do problema de classificação do *status* do gene MGMT são realizadas. Todas essas abordagens são apresentadas para introduzir e validar as técnicas exploradas no trabalho. Esse capítulo também traz informações sobre alguns dos conjuntos de dados disponíveis na literatura;
- no quarto capítulo toda a metodologia por trás das experimentações realizadas é apresentada. Essa metodologia inclui os conjuntos de dados utilizados, as técnicas utilizadas durante os treinamentos, métricas avaliadas e comparações realizadas;
- o quinto capítulo apresenta os resultados experimentais para todos os experimentos, além de discussões acerca desses experimentos. A partir da análise desses resultados os objetivos do trabalho podem ser alcançados;
- o sexto e último capítulo detalha as conclusões obtidas com o trabalho bem como as sugestões para trabalhos futuros.



## CAPÍTULO 2

---

### Conceitos básicos

---

Esse capítulo traz diferentes conceitos úteis para o entendimento do trabalho. Temas como aprendizado de máquina, redes neurais convolucionais, técnicas de avaliação de modelos e também técnicas de processamento de imagens médicas serão explorados.

#### 2.1 APRENDIZADO DE MÁQUINA

Aprendizado de máquina pode ser definido como a capacidade de uma máquina resolver determinado problema sem ser explicitamente programada a fazer isso. Essa capacidade de resolução vem do aprendizado obtido a partir de um treinamento baseado em dados de entrada. Matematicamente, a ideia principal envolve a minimização numérica de determinada função objetivo, chamada de função custo e relacionada ao problema a ser resolvido.

Os modelos são treinados com base em um conjunto de características (ou *features*) que são variáveis explicativas que descrevem o dado, utilizadas para inferir o valor da variável alvo (ou *target*). O conjunto de *features* não necessariamente precisa ser da forma tabular: diferentes categorias de dados não estruturados (imagens, áudios e textos) podem ser utilizados como características para treinamento de modelos de aprendizado de máquina.

A partir do conjunto de dados relacionado ao problema, um algoritmo de aprendizado de máquina é ajustado (ou treinado). Isso faz com que o processo por trás da geração da variável alvo seja aprendido/modelado. A partir do modelo ajustado, técnicas de avaliação de modelos de aprendizado de máquina são utilizadas para verificar a capacidade de generalização (gerar boas previsões em dados novos) do modelo.

Diferentes categorias de problema podem ser resolvidas a partir do uso de aprendizado de máquina. Soluções para detecção de fraude, identificação de problemas médicos, previsão de valores imobiliários já são totalmente desenvolvidas com algoritmos de *machine learning*.

Grande parte dos problemas resolvidos podem ser encaixados em duas áreas: aprendizado supervisionado e aprendizado não-supervisionado.

### 2.1.1 Aprendizado supervisionado

Problemas de aprendizado supervisionado envolvem a presença de uma variável alvo para treinamento do modelo. A partir da apresentação da *target* durante o treinamento o modelo vai se ajustando na tentativa de minimizar a função custo definida. Duas categorias de problemas podem ser encontrados nessa área: regressão e classificação.

Problemas de regressão geram uma saída numérica contínua. Esse tipo de problema costuma ser utilizada quando a variável a ser predita varia continuamente como o preço de uma ação, a temperatura de uma cidade, etc. Para ajuste do modelo um conjunto de *features* e uma variável alvo precisam ser utilizados.

A regressão linear é um dos principais modelo de regressão utilizados. É definida por meio da equação (1):

$$y = \mathbf{w}^T \mathbf{x} \quad (1)$$

onde  $\mathbf{w} = [w_0, w_1, \dots, w_n]$  são os parâmetros ajustados durante o treinamento e  $\mathbf{x} = [1, x_1, \dots, x_n]$  representam as *features* de entrada (WATT; BORHANI; KATSAGGELOS, 2020).

Devido à sua simplicidade, esse modelo se torna altamente interpretável, tendo cada coeficiente uma relação direta com a forma que determinada característica se relaciona com a variável-alvo.

Um exemplo de aplicação de problemas de regressão na área médica é na predição do tempo de sobrevivência de um paciente submetido a uma cirurgia para remoção de um tumor cerebral (SUN *et al.*, 2019). O conjunto de dados necessário para treinamento deve ser constituído pelo tempo de vida de cada paciente (que será a variável-alvo) e suas características, as quais podem ser informações sobre o paciente/tumor ou até mesmo imagens representando seu cérebro/tumor.

Diferentemente de problemas de regressão, a classificação envolve a predição de classes. Assim como na regressão linear, os dados utilizados para resolução dessa categoria de problema são as *features* que podem representar as características de um tumor cerebral e a variável alvo, que pode representar a sobrevivência de um paciente a cirurgia (uma variável do tipo sim/não). A quantidade de categorias em problemas de classificação pode ser variada, sendo comumente encontrados problemas de classificação binária (envolvem a predição de duas categoria).

Tipicamente duas saídas podem ser associadas a um modelo de classificação: um nível de confiança (*score*), definido como um valor real entre 0 e 1, que representa quão confiante o modelo está em relação à classificação de determinada classe; ou o valor da classe, sendo tipicamente obtido via aplicação de uma regra de decisão ao nível de confiança, baseada em um limiar definido a partir do problema.

Dois modelos podem ser destacados dentro dessa categoria: regressão logística e árvores de decisão. Esse destaque vem da sua versatilidade e aplicabilidade em diferentes problemas.

A regressão logística pode ser definida pela equação (2). Nota-se a similaridade com a equação (1). Pode-se verificar que a regressão logística se reduz à combinação de uma regressão linear com uma função na saída  $\sigma$ .

$$y = \sigma(\mathbf{w}^T \mathbf{x}) \quad (2)$$

Nessa equação  $w$  e  $x$  são definidos da mesma forma que na equação (1). O termo  $\sigma$  é conhecido como função sigmoide e definido pela equação (3).

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (3)$$

A função custo entropia cruzada, definida pela equação (4), geralmente é a utilizada nos treinamentos de modelos de classificação.

$$\mathcal{L}(y, \hat{y}) = \frac{1}{N} \sum_{n=1}^N (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \quad (4)$$

Nessa equação  $N$  é o número de amostras,  $y_i$  é o valor real da classe (sendo 0 ou 1 em um problema de classificação binária). O termo  $\hat{y}$  é a confiança de saída do modelo (saída da função sigmoide no caso da regressão logística).

Intuitivamente, a ideia dessa função custo é gerar penalidades elevadas caso a saída do modelo seja 1 (ou próximo) e a classe real seja 0 (ou vice-versa). Além de ser derivável, possibilitando sua resolução através de métodos numéricos.

O modelo baseado em árvores de decisão tenta reduzir o grau de desorganização de um conjunto de dados a partir da definição de limiares de classificação baseados nas variáveis de entrada do modelo. Esse grau de desorganização é medido pela entropia, definida pela equação:

$$H(p) = \sum_{i=1}^N -p_i \log_2(p_i) \quad (5)$$

A variável  $p_i$  é definida como a probabilidade de se obter a classe  $i$  no conjunto de dados. Os valores da entropia variam entre 0 e 1 sendo o valor de 1 o maior valor possível, indicando completa desorganização.

Um exemplo intuitivo de como uma árvore de decisão é criada pode ser obtido partindo-se do conjunto de idades de pessoas numa sala. Podemos atribuir a cada pessoa um rótulo 1 caso seja maior de idade e 0 caso contrário. Uma árvore de decisão ajustada nesses dados terá como saída uma entropia 0 com um valor de limiar para classificação igual a 18 anos. A Figura 2 representa esse conjunto de dados bem como a construção dessa árvore de decisão.

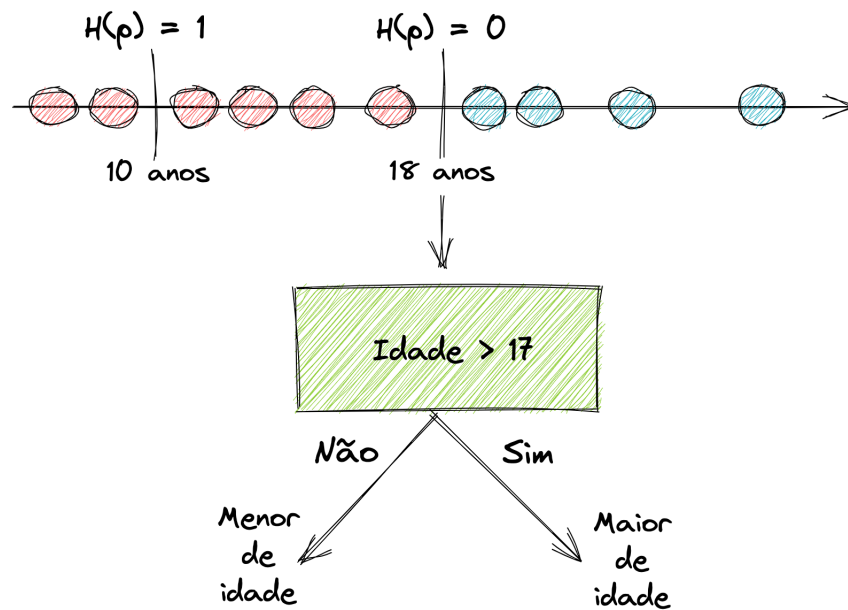


Figura 2 – Exemplificação da construção de uma árvore de decisão a partir de um conjunto de dados simples.

Devido a essa elevada explicabilidade, árvores de decisão acabam sendo utilizadas em diferentes aplicações para descrição de diferentes fenômenos. Sua implementação também pode ser vista como um conjunto de condições se/então, tornando sua análise intuitiva.

Árvores de decisão conseguem gerar fronteiras de decisão complexas, se adaptando muito bem a diferentes padrões nos dados, principalmente em problemas envolvendo um elevado nível de não-linearidades. Essa característica combinada com sua capacidade de explicação as tornam extremamente utilizados para solução de problemas de *machine learning*.

Com o intuito de melhorar o desempenho de modelos de classificação (ou de regressão), técnicas de *ensemble* (combinação) foram desenvolvidas. Essas técnicas se baseiam na ideia de combinar diferentes classificadores treinados de diferentes formas para se obter previsões mais corretas. Essa tentativa tenta simular um comitê de especialistas para obtenção da previsão. Diferentes técnicas de *ensemble* vêm sendo utilizados para aumentar o poder preditivo de modelos. Técnicas como *bagging*, *stacking* e *boosting* são aplicadas com o intuito de se obter maior capacidade de generalização em dados não vistos, sendo definidas da seguinte forma:

- O método de *bagging* se baseia na ideia de treinar simultaneamente modelos fracos (tipicamente árvores de decisão com profundidade reduzida) em diferentes partes dos dados. Isso gera um conjunto de modelos razoavelmente independentes, que podem ser utilizados em conjunto para realizar as previsões (GÉRON, 2019);
- o método de *stacking* utiliza a ideia de um *meta learner* para ajuste em duas etapas do modelo. Nessa técnica a saída de vários modelos é combinada a partir da utilização de outro modelo, treinado nos dados. Isso garante que a função de combinação seja a mais adequada àquele problema (GÉRON, 2019).

- por fim, a técnica de *boosting* se baseia na ideia de treinar diferentes classificadores fracos, sequencialmente, nos resíduos gerados entre as previsões. Esse tipo de treinamento consegue reduzir significativamente o nível de viés do modelo (GÉRON, 2019) porém acaba facilitando a ocorrência de *overfitting*.

Atualmente, devido à sua popularidade em competições de *machine learning*, a técnica de *boosting* vem tendo seu uso cada vez mais difundido em aplicações nas mais diferentes áreas (CHEN, T. *et al.*, 2015). A popularidade dessa técnica pode ser notada no sucesso de diferentes implementações de código aberto dessa técnica como *XGBoost* (CHEN, T. *et al.*, 2015), *CatBoost* (PROKHORENKOVA *et al.*, 2018) e *LightGBM* (KE *et al.*, 2017), cada uma compartilhando o mesmo conceito de *ensemble* de modelos fracos (mas com suas devidas particularidades).

A classificação pode ser generalizada para resolver problemas de segmentação semântica. Essa categoria de problema visa a geração de máscaras para segmentação de regiões em imagens de entrada. O processo de treinamento é realizado de forma que um modelo de aprendizado de máquina aprenda a classificar cada *pixel* individualmente para que, após a classificação de todos os *pixels*, a máscara de saída possa ser gerada.

Essa máscara pode ser utilizada para remoção dos *pixels* fora da região de interesse da imagem de entrada. A Figura 3 apresenta o resultado da aplicação de uma máscara a uma imagem de entrada.

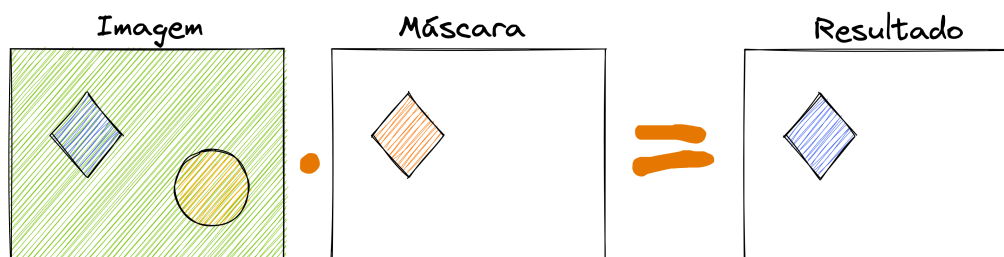


Figura 3 – Resultado da aplicação de uma máscara a uma imagem de entrada.

Modelos clássicos como regressão logística ou árvores de decisão acabam não sendo utilizados nessa tarefa devido a seu desempenho não atingir patamares satisfatórios, principalmente devido ao elevado número de parâmetros de entrada e de saídas que são inerentes a essa categoria de problema e a natureza não-estruturada dos dados.

### 2.1.2 Aprendizado não-supervisionado

Em problemas onde a variável alvo não existe ou até mesmo é muito difícil de ser obtida, métodos de aprendizado não-supervisionado podem ser aplicados. Essa área utiliza técnicas que buscam encontrar segmentos semelhantes (*clusters*) nos dados, a partir da análise das *features* do problema.

Os métodos de clusterização se baseiam na ideia de proximidade entre as amostras. A Figura 4 apresenta visualmente essa ideia. Um dos principais métodos é o *k-means*, que parte da minimização de uma função custo baseada em distância para determinação dos *clusters* e de seus membros.

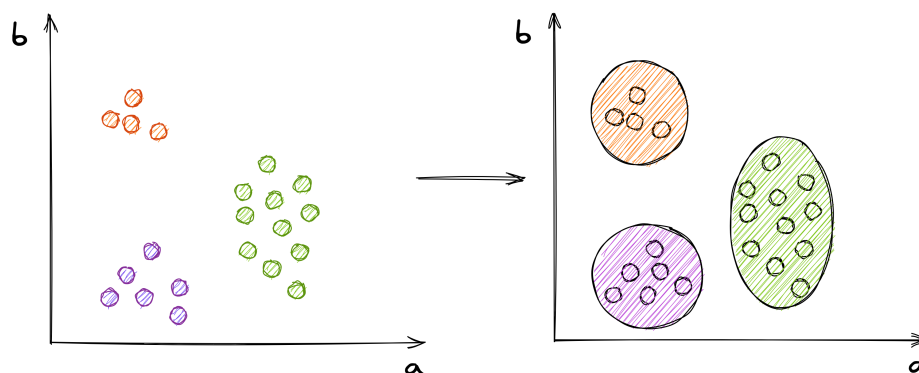


Figura 4 – A clusterização visa encontrar grupos semelhantes num espaço de *features* de entrada.

## 2.2 AVALIAÇÃO DE MODELOS DE CLASSIFICAÇÃO BINÁRIA

A partir dos modelos treinados em determinada tarefa, a avaliação de seu desempenho e capacidade de generalização em dados não vistos precisa ser determinada. Diferentes abordagens devem ser utilizadas para avaliação de diferentes tipos de problemas de *machine learning* devido a suas características distintas.

O passo comum para validação de modelos é dividir o conjunto de dados, de maneira aleatória, em duas partes: uma utilizada para treinamento dos algoritmos e outra utilizada para verificação de desempenho e generalização. Essa técnica de validação cruzada é conhecida como *hold-out*. Tipicamente o conjunto de treinamento é composto por 75% dos dados originais enquanto o restante é utilizado para o conjunto de teste (DUBITZKY; GRANZOW; BERRAR, 2007).

A divisão do conjunto de dados é utilizada para se inferir o grau de generalização do modelo em dados não vistos. Uma má separação dos dados pode acarretar resultados errôneos sobre a generalização obtida pela modelo, fazendo com que más decisões sejam tomadas. A avaliação também é útil para se determinar se, após o treinamento, o modelo não está sofrendo de dois problemas comumente encontrados:

- *overfitting*: quando o modelo consegue elevado desempenho no conjunto de treinamento, mas não consegue generalizar bem em um conjunto de dados não visto;
- *underfitting*: quando o modelo não obtém um desempenho razoável em seu treinamento.

Além da técnica de *hold-out*, técnicas de validação cruzada baseadas em *k-folds* podem ser aplicadas. Nesse caso o conjunto de dados é dividido repetidamente em *k* grupos, sendo 1 deles utilizado como validação e o restante *k* - 1 utilizado para treino (DUBITZKY; GRANZOW; BERRAR, 2007). Essa técnica permite obter um bom indicador do poder de generalização de um algoritmo a partir da análise das métricas obtidas a cada *fold*, além de ser útil quando há um baixo número de amostras disponíveis para treinamento de um modelo. A Figura 5 apresenta os passos que ocorrem durante a aplicação da validação cruzada de 5 *folds*.

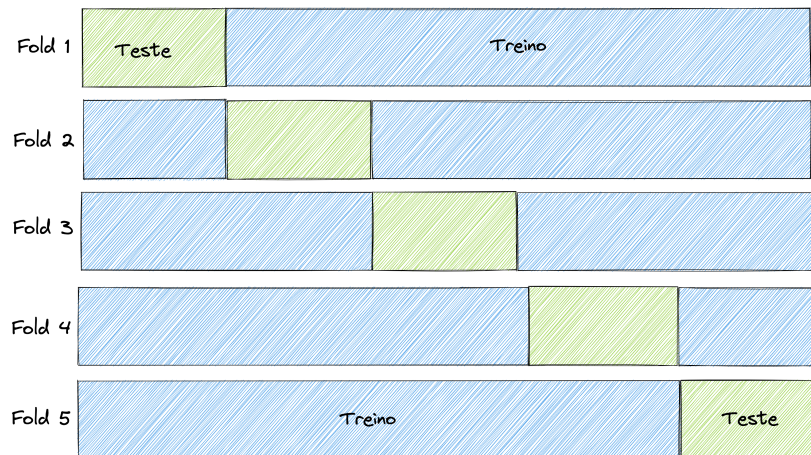


Figura 5 – Exemplo de validação cruzada de 5 *folds*.

A partir dos conjuntos (de treino e teste) os problemas de classificação binária podem ser avaliados com base em métricas calculadas com o valor do seu *score* de saída (o qual é um número real entre 0 e 1) ou com base em uma classe (muitas vezes gerada com base no *score* de saída).

Quando a análise é realizada em relação às classes geradas pelo modelo de classificação, algumas grandezas podem ser definidas (as classes serão representadas pelos valores 0 e 1):

- Verdadeiro positivo (VP): predição de classe 1 que realmente é da classe 1;
- Verdadeiro negativo (VN): predição de classe 0 que realmente é da classe 0;
- Falso positivo (FP): predição de classe 1 em amostra que é da classe 0;
- Falso negativo (FN): predição da classe 0 em amostra que é da classe 1.

Essas grandezas podem ser vistas qualitativamente em uma visualização conhecida como matriz de confusão apresentada na Figura 6.

A partir desses valores, diferentes métricas de classificação podem ser calculadas. Cada uma dessas métricas tem com objetivo quantificar uma informação diferente sobre as classificações geradas pelo modelo.

Duas métricas são comumente calculadas com base na matriz de confusão: a precisão (*precision*) é definida como  $\frac{VP}{VP+FP}$ ; e a revocação (*recall*) definida como  $\frac{VP}{VP+FN}$  (FAWCETT, 2006). Uma métrica que pode ser obtida a partir da média harmônica dessas grandezas é o *score* F1.



		classe real	
		1	0
Predição	1	VP	FP
	0	FN	VN

Figura 6 – Representação dos valores VP, VN, FP, FN em uma matriz de confusão.

Outra métrica, muito utilizada em problemas de classificação devido à sua simplicidade de entendimento é a acurácia, que mede quantas predições são corretas independentemente de sua classe. Essa métrica tem problemas principalmente quando usada em problemas desbalanceados, nos quais a quantidade de dados de determinada classe é muito superior a dados de outra.

A importância de cada uma dessas métricas é dependente do problema em análise. Por exemplo, determinados problemas de classificação de casos fraudulentos podem se beneficiar de modelos de classificação com baixo número de falsos positivos (elevada precisão) devido aos impactos que esses falsos positivos tem na operação do negócio. Por outro lado, sistemas de diagnóstico precoce de uma doença fatal podem se beneficiar de uma elevada revocação, para evitar que pacientes com a doença sejam ignorados.

Modelos de classificação tipicamente obtêm a classe de saída a partir da aplicação de uma função não-linear aplicada ao *score* de saída do modelo. A Equação (6) apresenta essa função, dependente de um limiar ( $\lambda$ ). Devido a isso, todas as métricas apresentadas são dependentes da escolha do limiar, o que torna essa escolha crucial para uma boa avaliação do problema.

$$f(x) = \begin{cases} 1, & \text{se } x > \lambda \\ 0, & \text{caso contrário} \end{cases} \quad (6)$$

Uma forma de se analisar a qualidade das predições geradas por um classificador binário, independentemente do limiar escolhido para classificação, é a análise de sua curva *receiver operating characteristic curve* (ROC). Essa curva é gerada a partir da variação do limiar de classificação utilizado para definir se determinado *score* deve gerar um valor pertencente a classe 0 ou pertencente classe 1. A partir das classes obtidas duas métricas são calculadas: taxa de falsos positivos ( $\frac{FP}{FP+VN}$ ) e taxa de verdadeiros positivos (revocação) (FAWCETT, 2006). Um exemplo de gráfico gerado pela combinação desses valores pode ser observado na



Figura 7. A reta pontilhada apresentada na Figura representa o desempenho de um classificador aleatório.

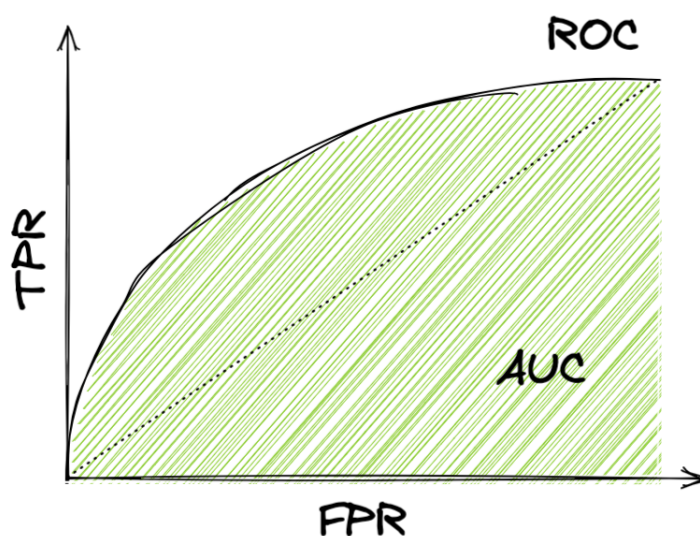


Figura 7 – Curva ROC de exemplo e reta identificando o desempenho aleatório.

Um dos problemas de se avaliar classificadores com a curva ROC é devido à natureza qualitativa e não quantitativa da análise. Uma forma de se analisar quantitativamente a curva ROC é a partir da área abaixo da curva. Essa área mede quão bem o modelo consegue discernir entre às duas classes de saída. Valores próximos a 1 indicam uma elevada capacidade de classificação enquanto valores próximos a 0,5 indicam um desempenho análogo ao de um classificador aleatório (representado pela reta  $f(x) = x$  na Figura 7).

As métricas calculadas em problemas de classificação binária podem ser estendidas para problemas de classificação semântica, já que, ao nível de *pixel*, esses problemas podem ser vistos como problemas de classificação.

Em problemas de segmentação semântica a principal grandeza a ser medida é quão próxima a máscara gerada é em relação à máscara de referência. A métrica utilizada para isso é chamada *Sørensen–Dice coefficient* (simplificada para coeficiente *dice*) e definida pela equação (7):

$$DSC = \frac{2|Y \cap \hat{Y}|}{|Y| + |\hat{Y}|} \quad (7)$$

na qual,  $Y$  é definido como a máscara original e  $\hat{Y}$  é definido como a máscara gerada.

Esse coeficiente varia entre 0 (máscara gerada e de referência totalmente discrepante) até 1 (concordância completa entre as máscaras). Em uma análise mais sutil pode-se verificar que esse coeficiente pode ser visto como a métrica *score F1* aplicada a problemas de segmentação semântica.

## 2.3 APRENDIZADO PROFUNDO

Aprendizado profundo (*deep learning*) é uma sub-área de *machine learning*. Essa área trabalha com modelos conhecidos como redes neurais artificiais e merece uma seção exclusiva devido à sua importância para o restante do trabalho e aplicabilidade na área médica.

### 2.3.1 Redes neurais densas

Redes neurais artificiais vêm sendo destaque nos últimos anos devido à sua aplicabilidade (com elevado desempenho) em tarefas de diferentes áreas. Esses modelos são comumente construídos com um elevado número de parâmetros, auxiliando na sua capacidade de aprendizado. Um neurônio (ou unidade) é o componente que constitui a rede neural sendo definido da seguinte forma:

$$\hat{y} = f(w^T x) \quad (8)$$

onde  $w$  é o vetor de pesos associado ao neurônio e  $x$  é o vetor de entrada. A função  $f$ , chamada função de ativação, pode ser qualquer função diferenciável e é responsável por inserir não linearidades no modelo.

Unidades são organizadas em camadas e redes neurais são constituídas por um conjunto de camadas. A saída de uma camada é a entrada da próxima. O termo *deep learning* vem da ideia de conectar diversas camadas, tornando a rede neural profunda. A Figura 8 representa uma rede neural densa, a qual tem esse nome devido a cada unidade de uma camada ser conectada a toda unidade da camada adjacente.

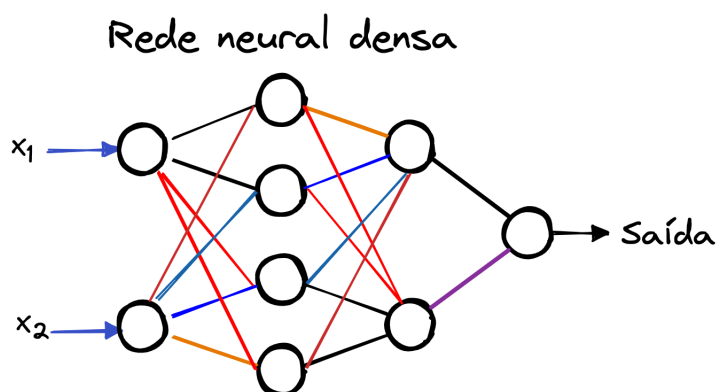


Figura 8 – Exemplo de rede neural densa com 2 entradas e 1 saída.

Uma forma interessante de se interpretar a conexão de camadas é analisá-la pela ótica de geração automática de características. Durante o treinamento, o modelo, de maneira autônoma, consegue determinar quais características e quais relações entre as características de entrada são as mais relevantes para resolução do problema em análise.

A quantidade de neurônios de saída (no caso da Figura 8 igual a 1) está diretamente ligada à quantidade de saídas do problema a ser modelado. Problemas de classificação multi-

classe (ou multi-rótulo) geralmente tem mais de uma saída enquanto problemas de classificação binária ou regressão (tipicamente) tem apenas uma unidade de saída.

A saída de uma unidade pode ou não passar por uma função de ativação. A principal utilidade dessa função é inserir não-linearidades úteis para melhorar a capacidade de aprendizado do modelo. Uma das principais funções custo utilizada é a *rectified linear unit* (RELU), representada pela equação (9).

$$f(x) = \begin{cases} x, & \text{se } x > 0 \\ 0, & \text{caso contrário} \end{cases} \quad (9)$$

essa função preserva características de funções lineares tornando o processo de treinamento mais simples (GOODFELLOW; BENGIO; COURVILLE, 2016).

Três funções são comumente utilizadas nas unidades de saída de uma rede neural:

- linear: função  $f(x) = x$ , tipicamente utilizada em problemas de regressão;
- sigmóide: análoga à equação (3), utilizada em problemas de classificação binária ou classificação multi-rótulo (nos quais a mesma entrada pode ser classificada em uma ou mais classes);
- *softmax*: função  $\sigma(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$  utilizada em problemas multi-classe (onde as classes são mutuamente exclusivas).

### 2.3.2 Redes neurais convolucionais

Variando a equação (8) que define a rede neural pode-se alterar a sua capacidade de resolução de determinadas tarefas bem como capacidade de treinamento. Redes neurais densas são uma categoria de modelo com dificuldades em lidar com imagens, devido a aspectos como: natureza do dado, informações contidas em vizinhança, variância a translação, além de necessitar de uma quantidade extremamente elevada de parâmetros.

Para problemas envolvendo imagens, uma nova camada foi desenvolvida, batizada de camada convolucional. Essa camada é constituída por parâmetros que geram saídas de acordo com grupos de informações (ou um conjunto de *pixels*), chamados de campos receptivos (GÉRON, 2019). Esses parâmetros são filtros que quando aplicados à imagem de entrada extraem determinadas características, aprendidas durante o treinamento. A Figura 9 exemplifica essa operação.

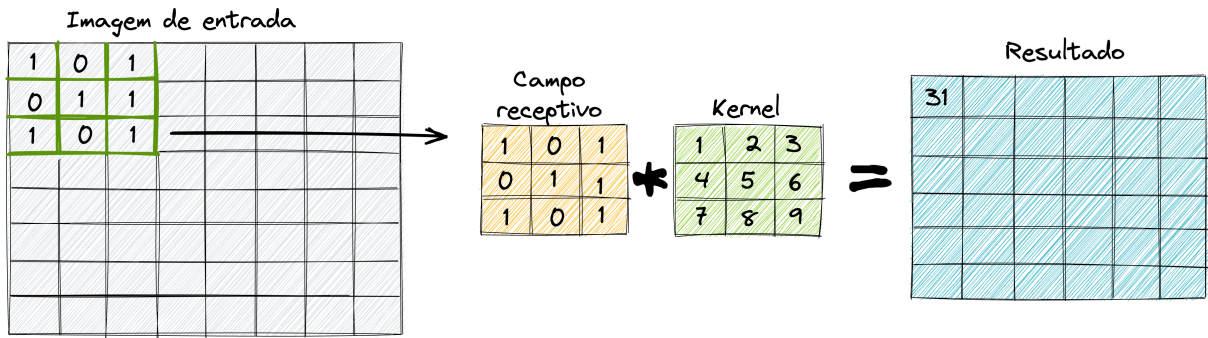


Figura 9 – Exemplo da operação de convolução entre uma imagem de entrada e um *kernel*.

Tanto a imagem de entrada como o filtro são representados por matrizes (ou, generalizando, tensores). Uma camada convolucional para duas dimensões pode ser definida pela equação (10):

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (10)$$

onde  $I$  é a matriz de entrada (ou imagem de entrada) e  $K$  é o filtro ou *kernel* (GOODFELLOW; BENGIO; COURVILLE, 2016).

Nota-se que a aplicação sucessiva de camadas convolucionais reduz o tamanho da entrada sucessivamente. Supondo um *kernel* de duas dimensões aplicado a um tensor quadrado de dimensão  $M$  o resultado da convolução entre esses elementos resultará em uma matriz quadrada de dimensão  $M - 1$ .

Para auxiliar a implementação prática de redes neurais convolucionais outras camadas foram desenvolvidas como a de *pooling*, a qual é tipicamente conectada à saída de uma camada convolucional. O intuito dela é reduzir as dimensões da imagem de entrada, mantendo a maior quantidade de informação possível. Isso ajuda a reduzir os requisitos de memória sem reduzir (significativamente) a quantidade de informação carregada na imagem (GÉRON, 2019).

Uma operação de *pooling* comumente usada é a *max pooling*, a qual consiste na operação *max* (retorna o valor máximo de um conjunto de números) aplicada a um conjunto de valores de entrada. A Figura 10 exemplifica o funcionamento dessa camada.

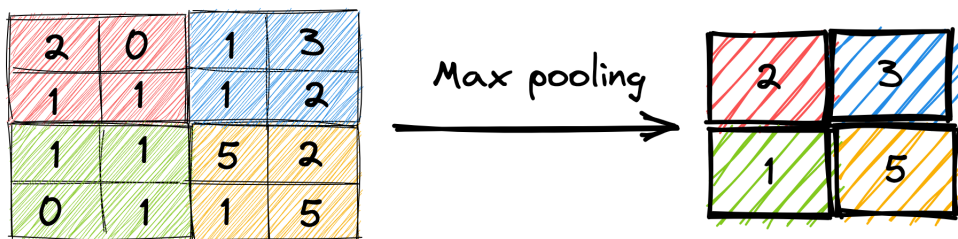


Figura 10 – Exemplificação do funcionamento da camada de *max pooling*.

Assim como antes, essas camadas podem ser conectadas entre si. Redes neurais convolucionais tendem a ser construídas com a composição de diferentes camadas convolucionais e de *pooling*, além de funções de ativação conectando essas camadas. A partir da conexão de camadas convolucionais com camadas de *pooling*, diferentes arquiteturas podem ser criadas, com capacidades de aprendizado distintas.

Em problemas de classificação de imagens, a camada final é necessariamente do tipo densa, seguindo a definição de neurônio apresentado na equação (8). A conexão entre as camadas convolucionais e a camada densa é feita através de uma operação que transforma o tensor de saída da camada convolucional em um tensor unidimensional.

A difusão das redes neurais convolucionais ocorreu a partir dos resultados obtidos na competição *ImageNet* de 2012. Nessa competição o vencedor foi, pela primeira vez, um modelo de redes neurais convolucionais batizado de *AlexNet*. O intuito era treinar um classificador em um conjunto de mais de 15 milhões de imagens pertencentes a mais de 20 mil categorias (DENG *et al.*, 2009).

A arquitetura da *AlexNet* é definida por camadas convolucionais seguidas por camadas de *max pooling* e funções de ativação do tipo RELU. As camadas finais são camadas densas e seguem a definição de neurônio apresentado na equação (8). Essa rede era composta por mais de 60 milhões de parâmetros. O resultado obtido por essa abordagem obteve uma taxa de erro de 15,3% na competição (para comparação, o segundo lugar obteve uma taxa de 26,2%) (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

A partir dessa competição, pesquisas começaram a desenvolver modelos cada vez mais profundos e a também desenvolver técnicas para lidar com o treinamento dessa categoria de modelo, devido a problemas que surgem a medida que camadas vão sendo adicionadas, como gradientes evanescentes (*vanishing gradients*). Esse problema ocorre em arquiteturas de elevada profundidade devido ao pequeno valor do gradiente propagado até as camadas iniciais, deteriorando seu processo de treinamento (BASODI *et al.*, 2020).

A arquitetura *ResNet* foi criada para resolver o problema da propagação do gradiente em redes neurais profundas. Nessa arquitetura caminhos residuais foram criados para reduzir ao máximo o problema dos gradientes evanescentes, construindo um trecho livre para propagação do gradiente ao longo da rede (HE, K. *et al.*, 2016). O bloco de construção dessa arquitetura pode ser observado na figura 11.

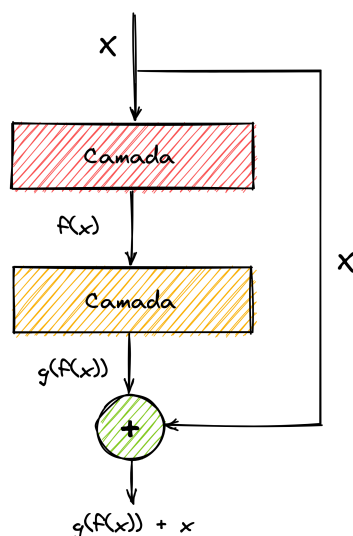


Figura 11 – Bloco construtivo da arquitetura ResNet.

Uma arquitetura mais recente, desenvolvida com foco na otimização computacional foi a *EfficientNet*. Seu desenvolvimento se baseou numa estratégia de otimização, a qual teve como saída a arquitetura mais eficiente para se atingir resultados considerados como o estado da arte em competições como a *ImageNet* (TAN; LE, Q., 2019).

As redes neurais convolucionais tiveram papel fundamental não só em problemas envolvendo classificação de imagens como também na evolução da área de segmentação semântica. Arquiteturas utilizadas para classificação de imagens tem aplicabilidade quase imediata nesse tipo de problema, devido à sua natureza ser muito próxima a de problemas de classificação comuns.

Um das principais diferenças entre problemas de classificação e problemas de segmentação semântica é a necessidade de utilização de camadas densas ao final. Em segmentação semântica as arquiteturas podem ser completamente convolucionais, devido à saída desses modelos ser diretamente a máscara desejada (uma imagem). Essa característica permite que imagens de diferentes tamanhos sejam processadas por modelos sem nenhum tipo de alteração em sua construção.

A abordagem mais natural de se resolver esse tipo de problema é construir uma rede neural convolucional, chamada *encoder*, para extrair características da imagem de entrada, além de reduzir suas dimensões. A partir desse conjunto de informações a máscara pode ser reconstruída, conectando a saída do *encoder* a outra rede neural convolucional chamada de *decoder*, conforme pode-se verificar na 12. Todos os blocos (azuis e vermelho) representam camadas convolucionais.

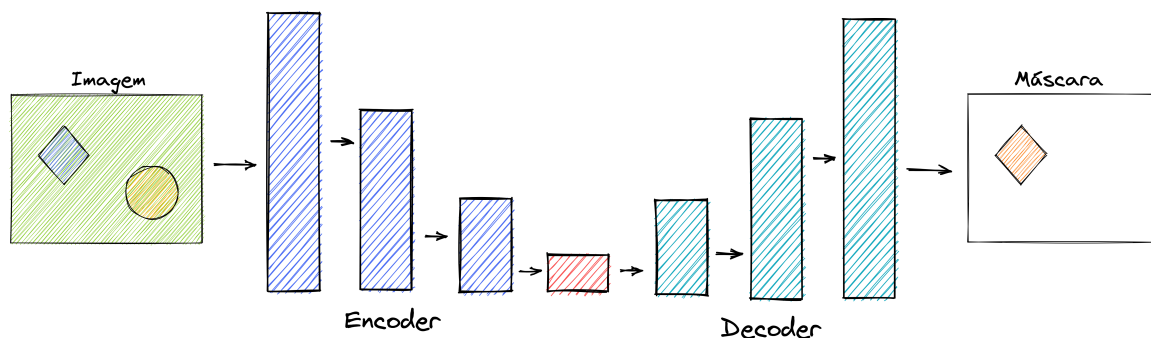


Figura 12 – Arquitetura *encoder-decoder* utilizada para segmentação semântica.

Para decodificar essas informações (e gerar a máscara de saída) o problema de aumentar as dimensões da imagem de entrada precisa ser resolvido devido à característica de redução de dimensões que as camadas convolucionais e de *pooling* tem. Métodos de *up-sampling* podem ser utilizados para essa tarefa. Um dos comumente utilizados é o método *nearest neighbor* que atua mantendo todos os valores dos *pixels* de determinada região iguais ao do *pixel* de entrada. Para elevar a capacidade de *up-sampling* a técnica conhecida como convolução transposta foi desenvolvida. Nesse método, a melhor forma de aumento da imagem é aprendida ao longo do treinamento para maximizar o desempenho na tarefa em questão. A operação de convolução transposta em um exemplo unidimensional pode ser observada na Figura 13.

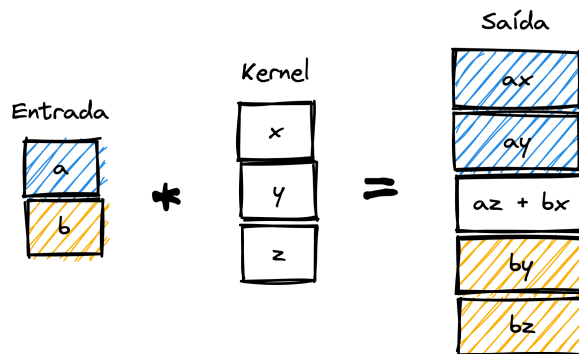


Figura 13 – Exemplo de convolução transposta.

### 2.3.3 Treinamento de redes neurais

A verdadeira utilidade de todas as arquiteturas de redes neurais surge após seu treinamento em determinada tarefa. Assim como em problemas de aprendizado de máquina clássicos, as redes neurais aprendem com base na minimização de uma função custo, atrelada ao problema de interesse.

As mesmas funções custo apresentadas nos problemas anteriores podem ser utilizadas para treinamento de redes neurais. Além disso, essa categoria de modelo é extremamente versátil, sendo muito fácil de adaptar funções custo para diferentes problemas, bem como construir diferentes arquiteturas de redes.

O treinamento de redes neurais é realizado por meio do algoritmo *backpropagation*, o qual se baseia no cálculo do gradiente da função custo para determinação dos pesos de cada unidade. Devido à complexidade e elevado número de parâmetros, essa minimização é realizada de maneira numérica, por meio do algoritmo do gradiente descendente, definido pela equação (11) (WATT; BORHANI; KATSAGGELOS, 2020):

$$\theta_{t+1} = \theta_t + \alpha \nabla J(\theta_t) \quad (11)$$

o parâmetro  $\alpha$  é conhecido como taxa de aprendizado e define o comportamento da atualização de pesos. Valores elevados podem levar a soluções divergentes (com o custo aumentando ao longo do treinamento). Em contrapartida, valores baixos podem levar a tempos de treinamento elevados. Na prática, a escolha do melhor valor depende da natureza do problema e dos dados, sendo esse ajuste realizado experimentalmente. O termo  $\nabla J(\theta_t)$  representa o gradiente da função custo, o qual é calculado para determinar a direção de atualização dos pesos da rede neural e a intensidade de atualização.

A atualização dos pesos é feita de maneira iterativa podendo ser realizada a cada amostra de dados ou a partir de lotes (*batches*) de dados. Em ambos os casos a minimização da função é ruidosa devido à amostragem presente. O treinamento baseado em *batches* é comumente aplicado devido à sua maior velocidade de convergência e capacidade de lidar com grandes volumes de dados. A equação (11) pode ser alterada para considerar fatores como o momento, que permite que soluções ótimas locais sejam evitadas.

Devido ao treinamento ser iterativo, um número máximo de iterações, chamado de número de épocas, deve ser definido. Uma época é completa quando todos os *batches* definidos para treinamento são utilizados para atualização dos pesos. Valores pequenos podem interromper o treinamento antes da completa minimização da função custo enquanto valores elevados podem causar *overfitting* durante o treinamento.

Para melhorar ainda mais o treinamento de modelos, algoritmos baseado em taxa de aprendizado adaptativa foram desenvolvidos (GOODFELLOW; BENGIO; COURVILLE, 2016): *AdaGrad*, *RMSProp* e *Adam* são alguns deles. O principal método de adaptação utilizado é o *Adam* o qual utiliza uma estimação do primeiro e segundo momentos do gradiente para atualização dos pesos (KINGMA; BA, 2014).

Para auxiliar no treinamento, algumas transformações podem ser aplicadas aos dados. Essas transformações podem ser camadas de normalização como a *batch normalization*, a qual faz com que cada *batch* tenha média nula e variância unitária. Essa camada pode ser utilizada durante o treinamento para ajudar a estabilizar o processo numérico.

Além de técnicas para melhorar a capacidade de aprendizado do modelo, técnicas de regularização são comumente aplicadas durante o treinamento desses modelos para evitar a ocorrência de *overfitting*. Regularizações clássicas (como penalidades L1 e L2) podem ser aplicadas à função custo, além de camadas conhecidas como *dropout*. Esse tipo de camada insere a capacidade de “desligamento” aleatório de unidades da rede neural durante o treinamento.



Isso faz com que todos os neurônios tenham que aprender informações relevantes da tarefa, não deixando nenhum com seus valores próximos ao inicial, o qual é definido aleatoriamente (SRIVASTAVA *et al.*, 2014). A Figura 14 ilustra a aplicação da técnica *dropout*.

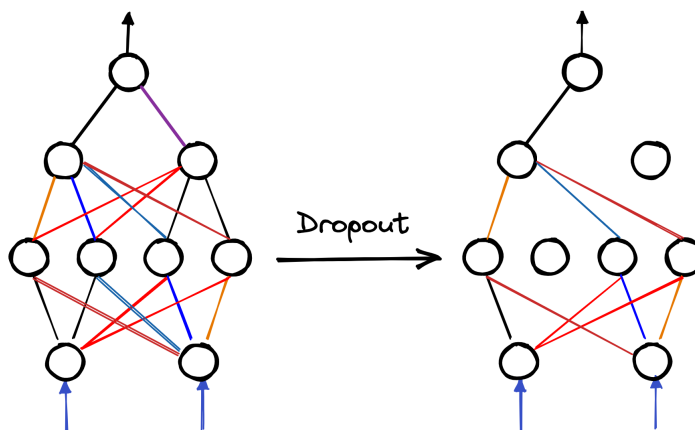


Figura 14 – Exemplo da aplicação de *dropout* em um modelo de *deep learning*.

*Data augmentation* é outra técnica utilizada para redução de *overfitting*. Nessa técnica, dados sintéticos são criados a partir de transformações aplicadas aos dados de treinamento. Essa técnica é tipicamente utilizada em problemas de classificação/segmentação de imagens devido à natureza invariante dessa categoria de dado a esse tipo de transformação. As transformações comumente utilizadas são rotações, inserções de ruído, *crop*, *shifts*, etc.

*Frameworks* de *deep learning* modernos (*PyTorch*, *Tensorflow*, etc) implementam de maneira muito eficiente todos os passos necessários para o desenvolvimento de redes neurais. Devido a seus mecanismos de diferenciação automática, esses *frameworks* são extremamente versáteis para construção de arquiteturas novas, com diferentes detalhes de implementação.

## 2.4 RESSONÂNCIAS MAGNÉTICAS

Os dados gerados durante diagnóstico de pacientes com tumores cerebrais podem ser imagens de ressonâncias magnéticas, obtidas durante exames médicos. Esse tipo de exame é o método de diagnóstico por imagem menos invasivo para tecidos moles (como o encéfalo), porém é o mais custoso em termos monetários (HAGE; IWASAKI, 2009) devido aos equipamentos utilizados, o que também dificulta a disponibilidade desse tipo de exame.

A criação dessas imagens é baseada na propriedade física de emissão de sinais de rádio de determinados núcleos de elementos, quando excitados a uma frequência chamada frequência de *Larmor*. Esse sinal de rádio pode ser captado e transformado em imagens (BLOCH, 1946). Esse exame não utiliza radiação ionizante, a qual pode afetar os tecidos, o que o torna seguro para uso médico.

Sequências de imagens com diferentes propriedades podem ser obtidas a partir de variações no exame. Quatro sequências de RMI são comumente criadas com base na variação

dos tempos de repetição (quantidade de tempo entre pulsos aplicados à mesma fatia) e tempos de eco (tempo entre o envio do pulso de radiofrequência e a recepção do sinal de eco). Além desses tempos, outros dois podem ser definidos:

- tempo de relaxamento T1: tempo necessário para que aproximadamente 63% dos prótons se alinhem com o campo magnético após o desligamento das ondas de rádio do exame;
- tempo de relaxamento T2: tempo necessário para que aproximadamente 37% dos prótons parem de girar sincronizadamente após o desligamento das ondas de rádio do exame.

Esses conceitos são úteis para definição das quatro categorias de sequências comumente encontradas em diferentes bases de dados públicas:

- nativo (T1): nesse tipo de sequência, gordura é apresentada de forma mais clara e água é apresentada de forma mais escura. Em relação aos tempos de repetição e de eco, esse tipo de sequência é produzido pela combinação curta (rápida) desses tempos;
- pós-contraste (T1-Gd): são sequências produzidas com a utilização de tempos de repetição e eco curtos e agentes de contraste, como gadolínio. Esse agente de contraste é utilizado pois, essa categoria de sequência gera sinais de alta intensidade para substâncias paramagnéticas;
- sequências ponderadas em T2 (T2): intensidade reduzida para gordura e para substâncias paramagnéticas. Essa sequência é útil para visualização de tumores, edemas, etc. Para sua produção, os tempos de repetição e de eco devem ser mais longos;
- recuperação de inversão atenuada por fluidos (FLAIR): semelhante a T2, com tempos de repetição e eco ainda mais longos. Essa sequência é especialmente útil em neuroimagem devido a sua capacidade de representar líquidos como o cefalorraquidiano.

## 2.5 PRÉ-PROCESSAMENTO DE RESSONÂNCIAS MAGNÉTICAS

Para garantir a usabilidade e padronização das imagens de ressonância magnética, algumas técnicas de pré-processamento acabam sendo comuns a todos que trabalham com esse tipo de problema. As principais técnicas utilizadas para aumentar a assertividade dos modelos de classificação e reduzir o ruído dos volumes para auxílio no diagnóstico médico podem ser agrupadas em quatro grandes grupos:

### 2.5.1 Co-registro de volumes

O co-registro dos volumes (ressonâncias magnéticas) visa padronizar tanto a orientação quanto a estrutura cerebral dos volumes a serem analisados (ZHANG *et al.*, 2019). Todos os volumes são registrados de acordo com um volume padrão (também chamado *atlas*).

Essa padronização auxilia tanto o profissional médico, reduzindo a variabilidade dos volumes que ele necessita analisar, quanto os sistemas de diagnóstico automático, que não precisam conseguir generalizar em um universo com uma grande variabilidade dos dados de entrada.

Para o co-registro, duas categorias de transformações são comumente aplicadas (ZHANG *et al.*, 2019):

- Transformações afim (lineares): visam reposicionar/reorientar o volume, tornando sua orientação mais próxima da orientação do volume padrão;
- Transformações elásticas (não lineares): visam alterar as estruturas cerebrais para deixá-las mais próximas da estrutura presente no volume padrão.

A Figura 15 apresenta uma fatia antes e depois a aplicação do co-registro baseado em uma transformação afim (rotação).

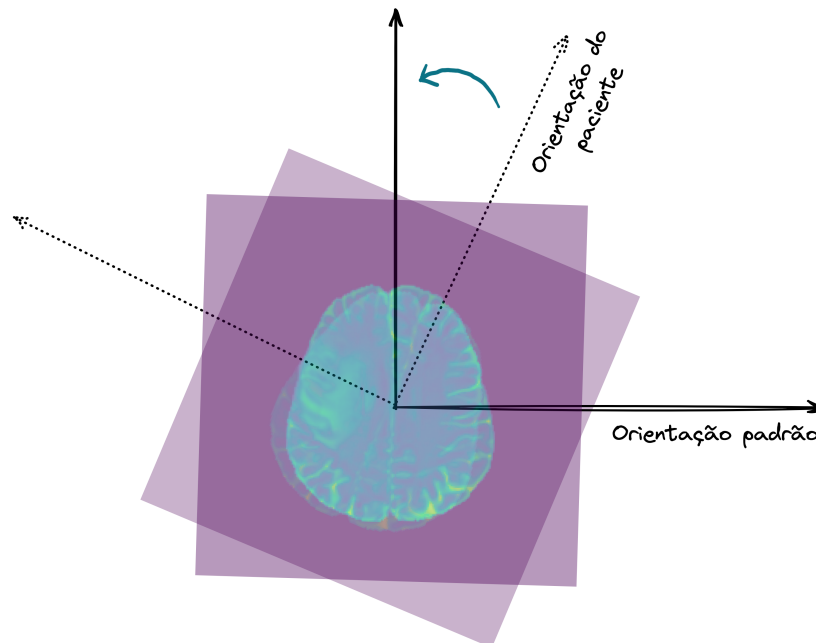


Figura 15 – Exemplo da aplicação do registro a uma RMI.

### 2.5.2 Remoção de crânio

Após o co-registro, uma etapa de remoção de crânio ou de qualquer estrutura fora do cérebro é realizada. Essa remoção visa reduzir a quantidade de informação irrelevante para o diagnóstico, tanto manual quanto automatizado, visto que algumas das estruturas

presentes no crânio podem se confundir com estruturas presentes nos tumores, principalmente em aspectos de coloração. Essa similaridade de coloração entre crânio e tumor tende a prejudicar o treinamento de modelos de redes neurais convolucionais, principalmente devido ao impacto nas etapas de normalização.

Normalmente duas abordagens para remoção de crânio são utilizadas: abordagens clássicas e baseadas em *machine learning*. Na abordagem clássica diferentes técnicas podem ser analisadas (FATIMA *et al.*, 2020):

- métodos baseados em superfície: nesses métodos um contorno é gerado ativamente para se moldar à superfície do cérebro em análise;
- operações morfológicas: são aplicadas operações de erosão e dilatação em conjunto com operações de detecção de borda;
- remoção de crânio baseada em co-registro: ligada ao co-registro, essa técnica registra o volume a um volume padrão com a segmentação do cérebro já existente e a partir disso a segmentação do volume novo é obtida.

Grande parte dos modelos de remoção de crânio baseados em *machine learning* funcionam com base no treinamento de um modelo de aprendizado supervisionado. Esse problema pode ser formulado como um problema de segmentação semântica, o qual tem como dados de entrada volumes brutos (cérebro com crânio e/ou outras estruturas fora do cérebro) e como saída a máscara do cérebro. Essa máscara, quando aplicada ao volume original, remove todas as estruturas exteriores ao cérebro.

Tanto na abordagem clássica como na abordagem via *machine learning* a qualidade da remoção de crânio gerada é obtida via comparação com volumes de referência. O coeficiente apresentado na equação (7) (métrica *dice*) é comumente utilizado para avaliação da qualidade de segmentações. Atualmente, as melhores abordagens são baseadas em redes neurais convolucionais, tais como as soluções apresentadas nos trabalhos de (KLEESIEK *et al.*, 2016), (IGLESIAS *et al.*, 2011) e de (ISENSEE *et al.*, 2019).

### 2.5.3 Segmentação de tumores cerebrais

Uma etapa de pré-processamento extremamente útil para auxílio no diagnóstico é a de segmentação de tumores. Essa etapa se beneficia dos resultados obtidos durante as etapas de registro e remoção de crânio.

Esse problema, assim como a remoção de crânio, pode ser abordado a partir de modelos de redes neurais convolucionais de segmentação semântica. Atualmente, essa categoria de arquitetura apresenta resultados que são o estado da arte em competições com foco na segmentação de tumores cerebrais a partir de imagens de ressonância magnética (BAID *et al.*, 2021).

A principal arquitetura utilizada para segmentação semântica é a U-Net (RONNEBERGER; FISCHER; BROX, 2015), a qual insere conexões (*skip connections*) entre as camadas

de *encoding* e as camadas de *decoding* do modelo de segmentação semântica apresentado na Figura 12. Essa conexão auxilia na propagação do gradiente durante o treinamento, melhorando o processo de aprendizado do modelo e a capacidade de segmentação de diferentes estruturas.

Algumas abordagens foram desenvolvidas para esse problema. Em (WANG, F. *et al.*, 2019) os autores se baseiam em uma arquitetura U-Net treinada com *patches* para obter resultados próximos ao estado da arte na tarefa. Já em (HAMGHALAM; LEI; WANG, T., 2019) os autores propõem a utilização de um modelo baseado em uma arquitetura *encoder-decoder*. Para otimização do desempenho e diminuição da possibilidade de *overfitting* do modelo os autores criaram um ramo de regularização baseado em um *variational autoencoder* que recria a imagem de entrada. Isso serviu para adicionar restrições nos pesos do modelo.

Atualmente o estado da arte é definido pelo artigo de (ISENSEE *et al.*, 2020), no qual os autores atingem um valor de métrica *dice* de 87,07%. Nesse trabalho é explorada a capacidade da arquitetura U-Net para desenvolvimento de um novo modelo chamado de NN-Unet utilizado para segmentação de tumores cerebrais. Esse artigo serve como porta de entrada para o trabalho (ISENSEE *et al.*, 2021), o qual define um *framework* de segmentação semântica de imagens médicas, que consegue trabalhar com segmentação de diferentes órgãos. Além de apresentar o modelo, as técnicas de pré-processamento são demonstradas e sua utilidade é comprovada por diferentes experimentos. O modelo NN-Unet pode ser observado na Figura 16.

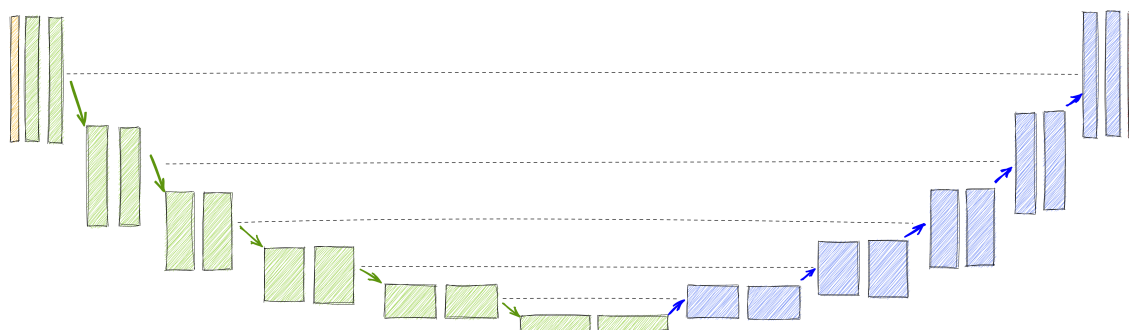


Figura 16 – Arquitetura NN-UNet utilizada para segmentação de tumores cerebrais.

Tanto os blocos verdes quanto os azuis representam a conexão de uma camada convolucional (3D) com uma camada de *instance normalization* seguidas por uma função de ativação RELU. As linhas tracejadas representam as *skip connections*. As setas verdes indicam uma operação de *pooling* e as setas azuis indicam uma operação de convolução transposta.

#### 2.5.4 Extração de *features* de tumores cerebrais

Uma etapa de pré-processamento especialmente útil para análises quantitativas de uma RMI é a extração de características numéricas relacionadas ao cérebro ou até mesmo ao tumor presente no cérebro de determinado paciente. Essas características são atributos estatísticos

que medem características do cérebro/tumor, permitindo a geração de um conjunto de dados estruturado a partir de imagens de ressonância magnética.

Esse conjunto de dados estruturado permite a construção de bases de dados úteis para análises estatísticas detalhadas da relação de cada atributo de um tumor cerebral com determinada característica genética, permitindo diagnósticos e até mesmo previsão de resposta a determinadas terapias (SINGH *et al.*, 2021).

Atualmente todo o processo de geração dessas *features* é automatizado por meio da biblioteca de código-aberto *PyRadiomics* (VAN GRIETHUYSEN *et al.*, 2017) a qual, a partir de um volume de entrada, consegue extrair mais de 100 características automaticamente. As características extraídas estão ligadas a mensurações de área, volume, níveis de energia, variação de intensidade, etc.

## CAPÍTULO 3

---

### Trabalhos relacionados

---

Nesta seção serão apresentados os trabalhos relacionados à pesquisa desenvolvida. Serão explorados os métodos utilizados pelos autores além de seus resultados e conclusões.

Diferentes abordagens para classificação do *status* do gene MGMT podem ser aplicadas, algumas mais simples e diretas e outras necessitando de etapas robustas tanto de pré-processamento quanto de pós-processamento. Essas abordagens podem ser agrupadas em 4 grupos:

- o primeiro grupo de problemas representa as abordagens realizadas por meio da extração de *features* radiômicas dos tumores dos pacientes e utilização de métodos de classificação clássicos para determinação do *status* do gene;
- o segundo grupo é formado por trabalhos que utilizam redes neurais convolucionais para classificação do *status* do gene MGMT a partir do exame completo do paciente (uma abordagem de classificação *end-to-end*);
- o terceiro grupo é formado por trabalhos que utilizam redes neurais convolucionais para classificação do *status* do gene MGMT a partir das fatias que compõe a RMI do paciente;
- o último grupo de trabalhos é composto por trabalhos que enquadram esse problema como um problema de segmentação semântica focado na classificação do *status* MGMT dos pacientes a partir de etapas de pós-processamento;

#### 3.1 CLASSIFICAÇÃO ATRAVÉS DE EXTRAÇÃO DE *FEATURES*

Alguns autores exploram diferentes problemas, como a previsão do tempo de vida após cirurgia, a partir de características radiômicas. A extração de características radiômicas de

tumores auxilia na identificação de correlação entre marcadores genéticos e características numéricas, auxiliando na identificação de padrões úteis para diagnósticos.

As características radiômicas são geradas a partir do volume de entrada do tumor combinado com diferentes algoritmos. Diferentes *features* se enquadram nesse padrão, como tamanho do tumor, *features* baseadas em localização, área superficial, tons de cinza, estatísticas de primeira e segunda ordem, etc (todas obtidas por meio da biblioteca *PyRadiomics*).

Seguindo essa abordagem, os autores do artigo (LE, N. Q. K. *et al.*, 2020) desenvolveram uma *pipeline* composta por uma etapa de extração de características do tumor (características radiômicas) seguida por um modelo de aprendizado de máquina para classificação do *status* do gene MGMT. Esse modelo foi desenvolvido com base em um conjunto de dados composto por 53 pacientes oriundos do *The Cancer Imaging Archive* (TCIA). Neste trabalho os autores reportam uma métrica ROC-AUC de 89,6% utilizando uma técnica de validação conhecida como *leave-one-out cross-validation* (DUBITZKY; GRANZOW; BERRAR, 2007), na qual o número de *folds* é igual ao número de amostras do conjunto de dados, sendo 1 amostra utilizada como conjunto de validação a cada iteração e o restante utilizado para treino. Essa abordagem de validação foi utilizada devido à pequena quantidade de pacientes presentes no conjunto de dados.

No trabalho (LE, N. Q. K. *et al.*, 2020) os autores realizam uma etapa de seleção de hiperparâmetros (para cada um dos modelos testados) baseada em uma técnica conhecida como *grid search*. Para seleção das melhores características os autores utilizaram uma técnica de eliminação recursiva de *features* (*recursive feature elimination*), o qual remove *features* de maneira iterativa. O modelo final selecionado foi o *XGBoost* (baseado na técnica *boosting*) foi o que obteve a melhor métrica ROC-AUC (média) ao longo da validação.

### 3.2 CLASSIFICAÇÃO END-TO-END COM DEEP LEARNING

Um dos métodos mais naturais de se resolver o problema de classificação do *status* do gene MGMT é a partir de um modelo de redes neurais convolucionais 3D treinado a partir do conjunto de exames e de rótulos. No trabalho (SAEED *et al.*, 2022) os autores exploram essa técnica a partir de testes com diferentes arquiteturas de redes neurais.

O conjunto de dados utilizado pelos autores é o oriundo da competição BRATS 2021, hospedada na plataforma *Kaggle*. Esse conjunto é composto por imagens de ressonância magnética de 577 pacientes com seus respectivos rótulos (MGMT metilado/MGMT não metilado) anotados por comitês de especialistas (BAID *et al.*, 2021).

Nesse trabalho a abordagem de classificação 3D baseada em uma arquitetura conhecida como *EfficientNet* obteve uma métrica ROC-AUC (%) de 64% no conjunto de teste, utilizando volumes de 128X128X128 para classificação. A função custo entropia cruzada binária foi utilizada para treinamento do modelo. Outras arquiteturas de redes neurais convolucionais foram testadas, todas tendo desempenho semelhante.

Para obtenção dos resultados uma busca de hiperparâmetros foi realizada visando



determinar número de épocas, necessidade de regularização, taxa de aprendizado e uso de *data augmentation*. Sempre com o intuito de maximizar a métrica ROC-AUC no conjunto de validação utilizado.

Os autores também realizam uma discussão acerca da qualidade de trabalhos publicados e dos conjuntos de dados utilizados, os quais, nos trabalhos analisados, apresentam pequena variabilidade devido à sua instituição médica de origem ser a mesma.

A partir dos resultados obtidos em (SAEED *et al.*, 2022), os autores questionam a viabilidade da classificação automática do *status* do gene MGMT a partir de imagens de ressonância magnética.

Além do trabalho (SAEED *et al.*, 2022), o repositório de projetos relacionados à competição BRATS 2021, hospedada na plataforma *Kaggle*, contém inúmeras soluções *end-to-end* 3D, as quais tem um desempenho semelhante ao reportado em (SAEED *et al.*, 2022).

### 3.3 DEEP LEARNING PARA CLASSIFICAÇÃO DE FATIAS

Em (KORFIATIS *et al.*, 2017) os autores exploram essa tarefa sob uma ótica de classificação de fatias por meio da utilização de uma arquitetura *ResNet*. Neste trabalho três variações dessa arquitetura são testadas: *ResNet18*, *ResNet34* e *ResNet50*, todas elas treinadas com base nas fatias que compõem os volumes de entrada. Os autores demonstram em seus resultados que redes neurais convolucionais são úteis para diagnóstico do *status* do gene MGMT, sendo a arquitetura *ResNet50* a melhor entre as três testadas, com uma acurácia de 94,90% (devido ao método utilizado pelos autores, a métrica ROC-AUC não pode ser computada neste trabalho).

Os autores utilizaram 155 exames de ressonância magnética, de pacientes acima de 18 anos, sendo 66 deles com gene MGMT metilado e 89 com gene MGMT não metilado. Todos os volumes continham as sequências T1 e T2 e eram oriundos de um conjunto de dados privado. O treinamento foi realizado a partir da extração das fatias (*slices*) que compõem o volume de entrada. As fatias foram classificadas individualmente com base em três classes: presença de metilação no gene, ausência de metilação no gene ou fatia sem tumor. O terceiro rótulo foi utilizado para auxiliar o treinamento do modelo, reduzindo o ruído dos dados.

O modelo foi treinado com uma camada final *softmax*, a qual gera rótulos mutuamente excludentes. Para se determinar o *status* de cada fatia, uma análise do maior valor de confiança foi realizada. Após o treinamento a inferência foi realizada e todas as fatias classificadas como sem tumor (fatias com maior nível de confiança nessa classe) foram removidas. Com as fatias restantes uma votação majoritária foi realizada para determinação do *status* do gene MGMT do paciente. Para selecionar a melhor arquitetura bem como os melhores hiperparâmetros, os autores utilizaram como estratégia a maximização da métrica F1 no conjunto de validação.

Em (CHANG *et al.*, 2018), a classificação de diferentes mutações genéticas é realizada. Os autores propõem identificar além da metilação do gene MGMT, o *status* do gene IDH e a co-deleção do gene 1p/19q, os quais são diferentes marcadores genéticos. A abordagem

utilizada pelos autores foi de uma classificação multiclases.

Os dados utilizados pelos autores foram obtidos na plataforma TCIA. Foram selecionados 259 pacientes para o estudo, todos os pacientes selecionados continham as quatro sequências de RMI: FLAIR, T2, T1 e T1-Gd, além da informação molecular acerca de seus marcadores genéticos. Os paciente selecionados não foram disponibilizados pelos autores.

Para gerar as fatias de entrada para treinamento os autores utilizaram a segmentação do tumor para extração de uma região de interesse (ROI) contendo o tumor. Após isso, essa ROI foi redimensionada para um tamanho de 32X32X4, onde 4 representa o número de canais presentes no volume de entrada (referentes às sequências da ressonância magnética). A arquitetura utilizada foi baseada em uma alteração da arquitetura *ResNet*. Camadas de normalização de *batch* (*batch normalization*) foram utilizadas, além de *dropout* para evitar o sobreajuste. Para geração da classificação final os autores utilizam a função *softmax* a qual gera um *score* referente a cada classe.

Para treino, técnicas de *data augmentation* foram utilizadas. As etapas selecionadas pelos autores foram deslocamentos aleatórios da imagem, remoção de canal de entrada e transformações afim, independentes em cada canal. As transformações afim resultaram em mudanças de escala, rotações e translações. *Adam* foi o otimizador selecionado pelos autores, com regularização L2.

A função custo utilizada para treinamento foi a entropia cruzada. Quando o custo no conjunto de validação atinge um platô, os autores utilizaram uma técnica de aumento do tamanho do *batch* para tentar retomar o processo de minimização do custo. Uma técnica de validação cruzada de 5 *folds* foi utilizada para obtenção dos resultados (em cada iteração 80% dos pacientes foram utilizados para treino e 20% para validação das métricas obtidas) para validar o desempenho do algoritmo desenvolvido.

Para obtenção dos *scores* de cada pacientes o valor médio de cada uma das saídas do modelo no conjunto formado pelas predições de todas as fatias de determinado paciente foi calculado. Caso o valor médio fosse superior a determinado limiar (definido experimentalmente) o paciente recebeu diagnóstico positivo para essa classe.

Em relação aos resultados obtidos, os autores reportaram uma acurácia (média ao longo dos 5 *folds*) de 94% na classificação da mutação IDH, 92% na identificação da co-deleção do gene 1p/19q e 81% na identificação da metilação do gene MGMT.

Em ambos os trabalhos, essa abordagem baseada em fatias aumenta a quantidade de dados utilizada para treinamento, em relação à técnicas baseadas em extração de *features* ou técnicas *end-to-end*, nas quais a quantidade de dados está limitada à quantidade de exames presentes, o que é útil em um cenário com poucos dados disponíveis.

### 3.4 ABORDAGEM VIA SEGMENTAÇÃO SEMÂNTICA

Um trabalho que explora essa forma de identificação é o trabalho (YOGANANDA *et al.*, 2021), onde os autores utilizam modelos de redes neurais para realizar tanto a segmentação do

tumor cerebral quanto o diagnóstico da metilação do gene MGMT. Neste trabalho os autores conseguiram obter uma ROC-AUC na detecção do *status* do tumor cerebral de 93,00% e uma métrica *dice* (para segmentação do tumor) de 82,00%. Todos os resultados foram obtidos por meio de uma estratégia de validação cruzada com 3 *folds*. Além da ROC-AUC e da métrica *dice*, os autores também reportaram valores de acurácia de 94,73%, com uma sensibilidade de 96,31% e uma especificidade de 91,66%, resultados promissores para essa tarefa. Foram selecionados 247 pacientes para desenvolvimento do trabalho, com seus dados oriundos do *The Image Cancer Archive* (TCIA).

A abordagem utilizada pelos autores visa realizar uma classificação do *status* do gene MGMT *voxel a voxel*. Com isso, a quantidade de informação processada pela rede neural se torna muito maior do que nas abordagens clássicas, onde apenas um *label* era atribuído a cada volume de entrada. Para realizar essa tarefa a máscara utilizada para segmentação é dividida em duas, uma para quando o *status* do gene for 0 e outra para quando o *status* do gene for 1. Isso faz com que o modelo de segmentação semântica seja treinado para gerar duas máscaras de saída, uma para pacientes com metilação do gene MGMT e outra para pacientes sem. Para treinamento essas máscaras são mutuamente exclusivas para cada paciente, já que o *label* associado a cada um deles é sempre o mesmo. A Figura 17 apresenta um esquema dessa arquitetura.

Para treinamento, os autores utilizam uma abordagem baseada em *patches*, a qual quebra o volume de entrada em pequenos volumes de 32X32X32. Além disso, essa geração de *patches* é realizada com um *overlap* de 75%, e mantém somente os *patches* com pelo menos 1 *voxel* de tumor presente o que, além de reduzir a quantidade de dados gerada, evita ruídos durante o treinamento. Os volumes de entrada utilizados são os referentes ao canal T2 somente, sendo que os autores realizaram testes utilizando todos os canais, mas não conseguiram identificar nenhum ganho aparente com essa utilização.

O treinamento apresentado em (YOGANANDA *et al.*, 2021) foi realizado a partir de um modelo pré-treinado na tarefa de classificação do *status* do gene IDH (BANGALORE YOGANANDA *et al.*, 2020) (o qual também foi treinado com dados oriundos do TCIA). Devido à validação cruzada de 3 *folds* ser utilizada na tarefa de classificação do *status* do gene IDH, os autores selecionam o melhor dos três modelos para inicialização dos pesos na tarefa de classificação do *status* do gene MGMT. Para treino o autor utiliza a mesma metodologia em ambos os trabalhos.

Após o treinamento, uma etapa de pós-processamento é realizada visando a reconstrução do volume original. Essa reconstrução visa gerar a máscara do tumor (utilizada para cálculo da métrica *dice*) além de gerar o grau de confiança (*score*) de saída de cada paciente. A reconstrução do volume é realizada por meio da fusão dos dois volumes gerados, por meio da maior componente conectada, e a geração do nível de confiança é gerada por meio da votação majoritária: qual o percentual de *voxels* classificados como MGMT metilado em relação ao total de *voxels*. Como limiar para geração das máscaras os autores utilizam o valor de 0,5.

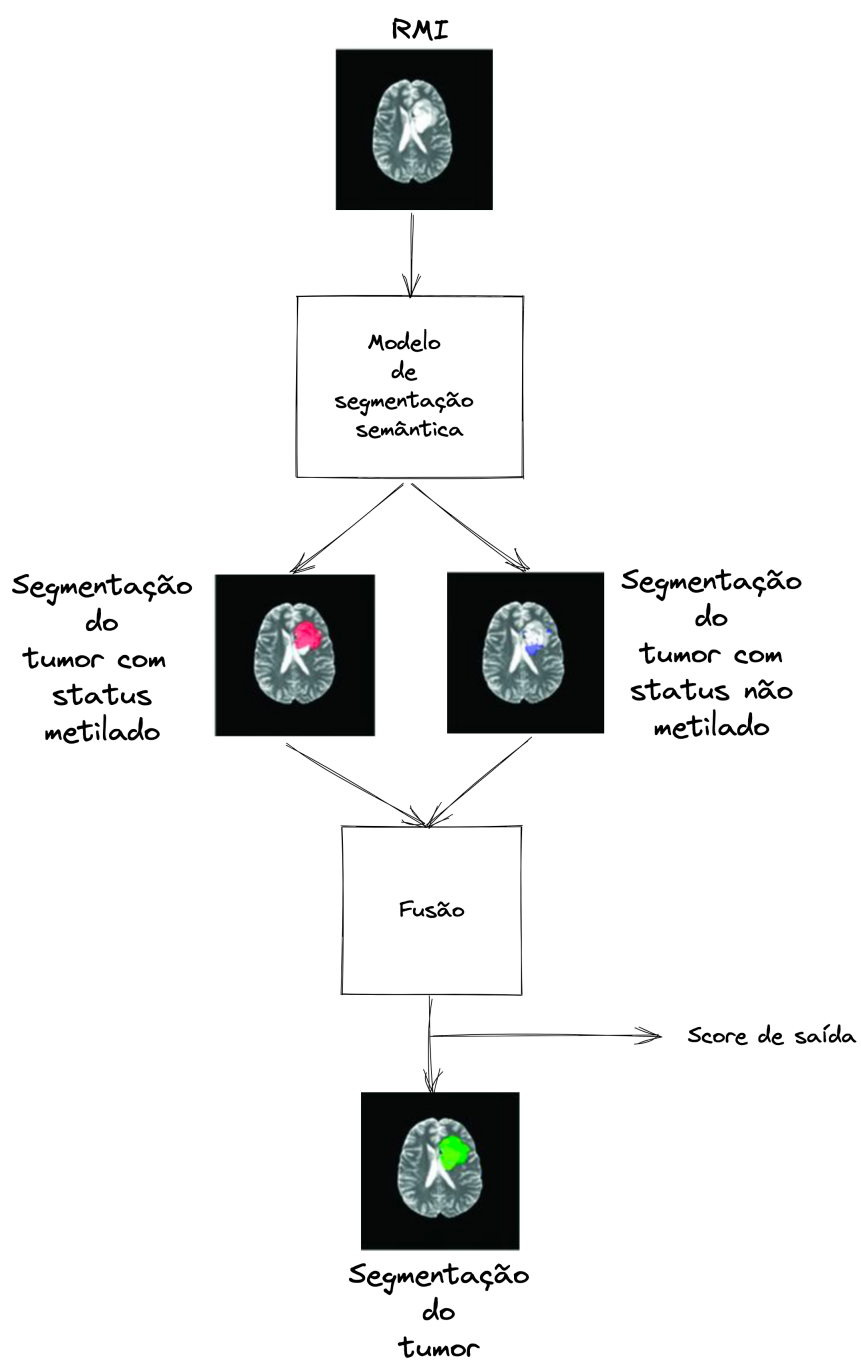


Figura 17 – Ilustração da abordagem baseada em segmentação semântica.

## CAPÍTULO 4

---

### Metodologia

---

Nesse capítulo será detalhada toda a metodologia utilizada no trabalho. Isso engloba informações sobre os dados utilizados e também sobre como os experimentos serão realizados.

#### 4.1 CONJUNTOS DE DADOS

O conjunto de dados principal do trabalho é o disponibilizado pela competição BRATS 2021. Esse conjunto traz volumes de ressonância magnética de diferentes pacientes, totalmente pré-processados. Além de ser o principal conjunto de dados, ele também é o maior conjunto de dados públicos disponível. Sua atualização é realizada anualmente, às vezes trazendo novos pacientes e novas tarefas possíveis de serem resolvidas com os dados disponibilizados.

Como pré-processamento, as imagens brutas (RMI) são submetidas a diferentes etapas pelos organizadores da competição: conversão dos arquivos DICOM para o format NIFTI, co-registro para o *template* SRI24, *resample* para uma resolução de  $1\text{mm}^3$  e remoção de crânio. Esse pré-processamento ajuda a reduzir a quantidade de ruído presente nas imagens e são passos comuns em todas as competições. Após essa etapa os volumes gerados estão padronizados e com um tamanho de 240X240X155.

Na última atualização (2021), as tarefas de segmentação semântica de tumores cerebrais e de classificação do *status* do gene MGMT dos pacientes com tumores cerebrais foram selecionadas para exploração. Em ambos os casos, o conjunto de dados trouxe rótulos anotados por comitês de especialistas, ajudando a garantir a qualidade do conjunto e a utilidade dos estudos desenvolvidos. Para a tarefa de classificação da metilação 577 exames foram disponibilizados, bem como seus rótulos (MGMT metilado/não metilado).

Na Figura 18 pode-se verificar um exemplo de fatia presente no conjunto de dados, além da máscara do tumor localizado nessas fatias. Cada fatia corresponde a uma das 4 sequências

de entrada apresentadas anteriormente. Além das fatias de entrada a Figura apresenta a segmentação do tumor presente nesse paciente, um exemplo de dado que é disponibilizado para a tarefa de segmentação semântica de tumores cerebrais.

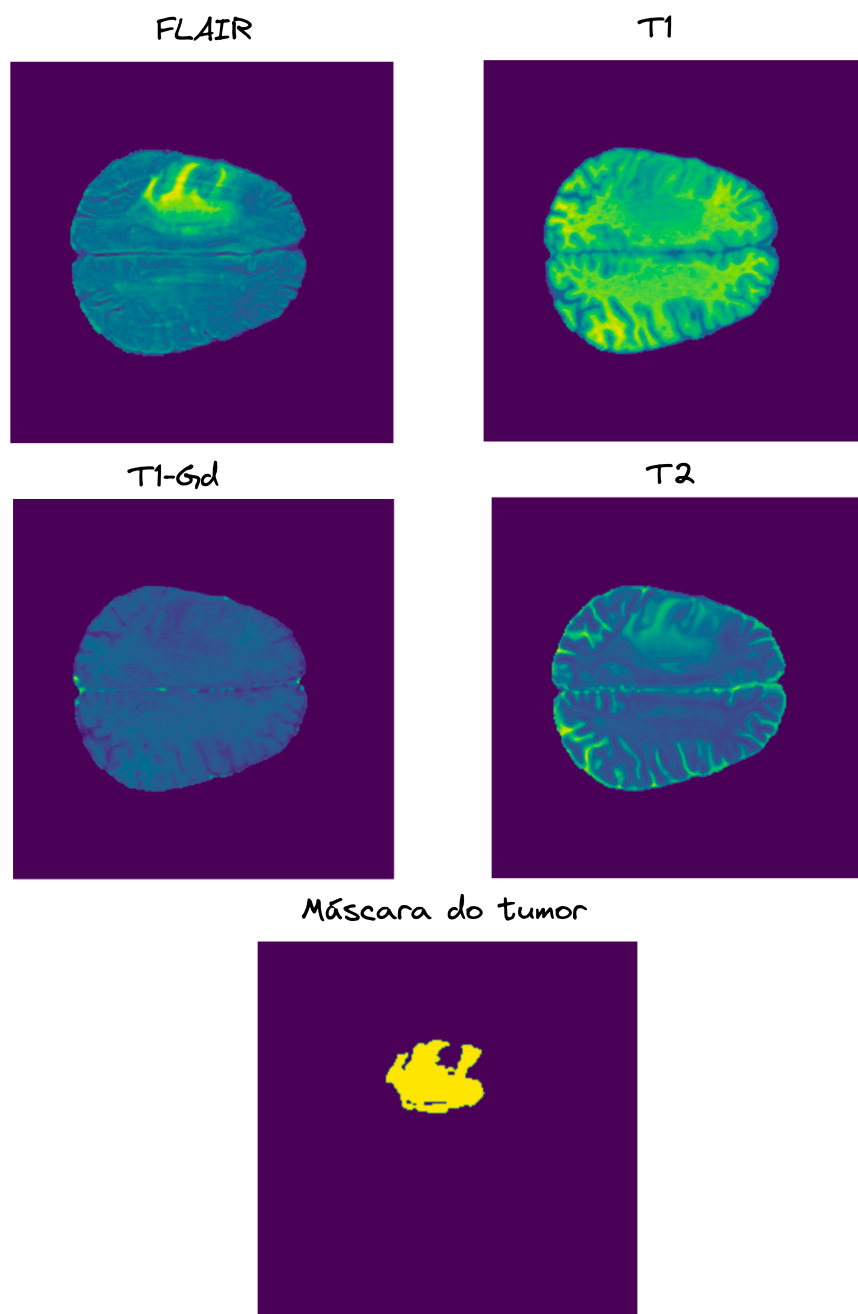


Figura 18 – Exemplo de sequências que compõe o conjunto de dados BRATS 2021.

Dos 577 pacientes presentes na base de dados, 52% são da classe MGMT metilado e 48% são da classe MGMT não metilado, mostrando um bom nível de balanceamento do conjunto de dados.

O segundo conjunto de dados é o mesmo conjunto utilizado no trabalho de (YOGANANDA *et al.*, 2021). Nesse trabalho o autor seleciona 247 pacientes oriundos do projeto

TCIA. Com seu trabalho, os autores disponibilizaram uma lista contendo os identificadores de cada paciente bem como seus rótulos referentes à classificação do *status* do gene MGMT e classificação do *status* do gene IDH. Esse conjunto foi selecionado com o intuito de se replicar os resultados apresentados pelos autores, tanto para se identificar o impacto de um pré-treinamento quanto para se verificar o impacto de um conjunto de dados diferente nos resultados obtidos.

Os autores não disponibilizaram qual sequência foi utilizada para cada paciente. Devido a isso, esse processo teve de ser realizado manualmente para cada um dos 247 pacientes presentes no conjunto de dados. Mesmo com análise manual, possíveis diferenças podem existir entre o conjunto de dados obtido e o utilizado pelos autores.

A partir das imagens selecionados, as seguintes etapas de pré-processamento foram aplicadas aos volumes brutos, gerando como resultado tanto um volume processado, de dimensões 240X240X155, quanto a segmentação do tumor presente no volume:

- registro dos volumes T1 para o *template* de SRI24;
- registro do volume T2 a partir do volume T1 previamente registrado;
- re-amostragem da imagem para o 1mm<sup>3</sup>;
- remoção do crânio e demais estruturas externas ao cérebro com a utilização do método HD-BET;
- segmentação de tumor com a utilização da arquitetura NN-UNet.

Alguns dos exames presentes nesse conjunto também estão presentes no conjunto BRATS 2021 (são um subconjunto do BRATS 2021). Devido a isso pode-se comparar a diferença entre os volumes pré-processados para se verificar a qualidade dos passos aplicados em relação a um padrão conhecido.

#### 4.1.1 Divisão do conjunto de dados

Para construção dos experimentos o conjunto de dados BRATS 2021 foi dividido em treino e teste (validação *hold-out*). Para treinamento, 75% do conjunto de dados foi amostrado e o restante dos dados foram utilizados como conjunto de testes. Uma amostragem estratificada foi utilizada na geração dos conjuntos para manter a distribuição original das classes. Em termos numéricos, 433 pacientes foram selecionados para treino enquanto 144 pacientes foram selecionadas para teste.

Os conjuntos de dados são divididos com base nos identificadores únicos dos pacientes. Essa abordagem é necessária para evitar que informações sobre pacientes utilizados no treinamento também sejam utilizadas para teste. Esse cenário pode ocorrer nas abordagens baseadas em fatias e em segmentação semântica, devido à relação 1:N entre pacientes e fatias/*patches*.

Todos os experimentos serão baseados no conjunto de treino e todas as comparações entre abordagens serão realizadas a partir do conjunto de teste, garantindo a capacidade de comparação entre os diferentes experimentos.

O conjunto de dados TCIA será utilizado para pré-treinamento na tarefa de classificação do gene IDH. Após isso, a rede pré-treinada será utilizada para treinamento na tarefa de classificação do MGMT tanto no conjunto de dados BRATS 2021 quanto no conjunto TCIA.

Os experimentos utilizando o conjunto de dados TCIA utilizaram a mesma divisão (75% dos dados para treinamento e 25% dos dados para teste). A seleção dos pacientes foi realizada para manter o conjunto de teste com os mesmos pacientes em ambas as tarefas (classificação do IDH e classificação do MGMT). Além disso, os pacientes oriundos do conjunto de dados BRATS 2021, selecionados para o conjunto de teste, também serão utilizados como conjunto de teste para os dados do TCIA, mantendo a consistência entre os conjunto de dados e evitando o problema de *data leakage*.

## 4.2 ABORDAGENS MAPEADAS

Nesse trabalho estão sendo investigadas diferentes técnicas de aprendizado de máquina para determinar a melhor abordagem em tarefas de classificação do *status* do gene MGMT de tumores cerebrais, com base nos trabalhos apresentados na seção de revisão bibliográfica.

Em todas as abordagens os experimentos serão realizados com base na sequência T2, a qual aparenta ser a mais utilizada para esse tipo de tarefa (experimentos com variação da sequência acabam extrapolando o tempo utilizado para desenvolvimento de cada abordagem).

A primeira abordagem testada é a baseada em extração de *features* (LE, N. Q. K. *et al.*, 2020). Nessa abordagem um modelo baseado em *machine learning* clássico será testado, com o intuito de se verificar a qualidade dessa técnica nessa categoria de problema.

A segunda abordagem testada é a mais natural para esse problema. Nessa abordagem um modelo de classificação 3D será treinado para classificar o *status* do gene MGMT a partir de imagens de ressonância magnética. Essa abordagem foi amplamente testada durante a competição BRATS 2021.

Seguindo a abordagem de (KORFIATIS *et al.*, 2017), de classificação de fatias baseada em *deep learning*, o modelo terá como entrada as fatias que compõe cada exame, sendo cada fatia classificada como: gene MGMT metilado, gene MGMT não metilado ou sem tumor para posterior aplicação de uma etapa de pós-processamento para classificação de cada paciente.

Por fim, a abordagem de (YOGANANDA *et al.*, 2021) será replicada no novo conjunto de dados. Os pré-processamento utilizado pelos autores será replicado (construção de *patches* de tamanho 32X32X32) para treinamento de um modelo de segmentação semântica e posterior classificação dos pacientes. Essa abordagem será realizada a partir de uma rede pré-treinada e também sem utilizar o pré-treino, para identificação do impacto dessa etapa nos resultados obtidos. Essa abordagem também será aplicada no conjunto de dados TCIA, visando a identificação do impacto do conjunto de dados nos resultados obtidos.

A ideia central dos experimentos é conseguir medir como o desempenho na tarefa de classificação varia conforme a abordagem utilizada. O desempenho será medido a partir da métrica ROC-AUC no conjunto de teste previamente separado. Dois motivos levaram à seleção



dessa métrica: consegue medir a capacidade de um classificador sem necessidade de um limiar de classificação e também devido à sua utilização na competição BRATS 2021, proporcionando uma elevada base de dados para comparação dos resultados.

#### 4.2.1 Abordagem baseada em extração de *features*

A abordagem baseada em extração de *features* utiliza a biblioteca *PyRadiomics* para construir um conjunto de características a partir dos volumes dos tumores. Para garantir que os resultados não sejam afetados pela qualidade das segmentações, somente as máscaras originais dos tumores foram utilizadas.

Todas as características são geradas a partir da segmentação do tumor realizada na sequência T2. A Figura 19 representa os passos dessa abordagem, a qual tem como principal etapa a extração de *features* radiômicas.

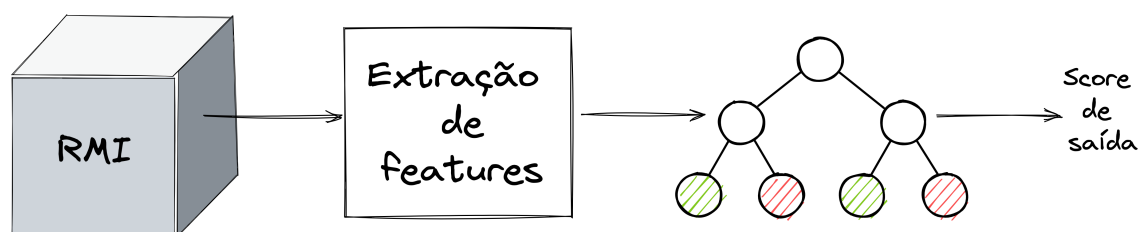


Figura 19 – Pipeline utilizada na abordagem baseada em extração de *features*

A partir do conjunto de características extraído, diferentes modelos serão testados: regressão logística, modelos baseados em *random forests* e *gradient boosting*. A determinação do melhor modelo será realizada por validação cruzada baseada no método *repeated random sub-sampling validation* (validação cruzada de Monte Carlo) (DUBITZKY; GRANZOW; BERRAR, 2007). Essa técnica é especialmente útil para determinação de métricas robustas em cenários de conjuntos de dados pequenos, porém é um método de validação computacionalmente intensivo.

O esquema de validação é apresentado na Figura 20. Nesse esquema, o número de simulações é independente da quantidade de dados, uma vantagem em relação à técnica de validação cruzada baseada em *k-fold*. A cada simulação uma nova seleção aleatória é feita no conjunto de dados original. Ao longo das simulações a métrica de teste também é calculada para cada um dos modelos treinados (sempre no mesmo conjunto de dados), possibilitando a construção de distribuições numéricas dessa métrica.

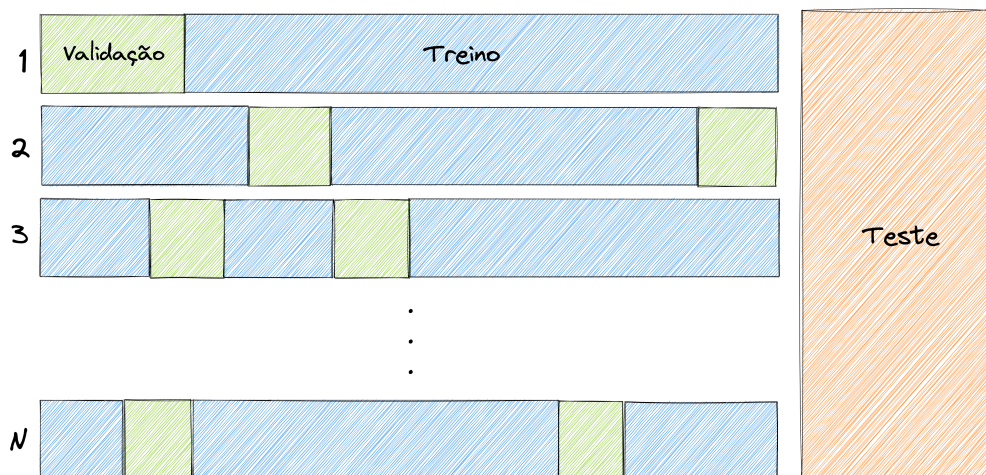


Figura 20 – Exemplificação do esquema de validação utilizado nessa abordagem.

#### 4.2.2 Abordagem 3D (*end-to-end*)

Na abordagem 3D os testes focaram em duas arquiteturas: *ResNet* e *EfficientNet*. Originalmente essas arquiteturas foram desenvolvidas para lidar com imagens, mas com devidas adaptações (como utilização de camadas convolucionais 3D), conseguem também lidar com volumes 3D.

Para treinamento e seleção de hiperparâmetros o conjunto de treino original será dividido novamente para construção de dois novos, sendo um utilizado para o treinamento e outro utilizado para validação em etapas de otimização de hiperparâmetros. O novo conjunto de treino é composto por 75% dos pacientes presentes no conjunto de treino original. O restante dos pacientes constitui o conjunto de validação. Essa divisão será utilizada em todas as abordagens envolvendo redes neurais convolucionais devido ao tempo de treinamento desses modelos.

Todos os volumes utilizados foram redimensionados de 240X240X155 para 128X128X128 para evitar erros de memória da GPU durante o treinamento, além de manter o tamanho de cada eixo igual. Caso necessário, para aumento do tamanho do *batch*, novos redimensionamentos podem ser realizados ao longo dos experimentos.

A função custo entropia cruzada binária será utilizada para definir os pesos da rede durante o treinamento. A saída da rede será um único *score*, gerado por uma função sigmoide. Devido a essa abordagem gerar um *score* de saída para cada paciente a métrica ROC-AUC poderá ser avaliada a cada época, sem a necessidade de uma etapa de pós-processamento.

Os hiperparâmetros serão ajustados à medida que for necessário, sempre com o intuito de se maximizar a métrica ROC-AUC no conjunto de validação. Para isso hiperparâmetros como taxa de aprendizado, tamanho de *batch*, número de épocas e regularização serão testados, além de técnicas de *data augmentation* para aumento da variabilidade do conjunto de dados (caso seja necessário).

A Figura 21 apresenta um diagrama dessa abordagem. Devido à capacidade de lidar com o exame completo, essa técnica requer a menor quantidade de etapas de pré/pós-

processamentos.

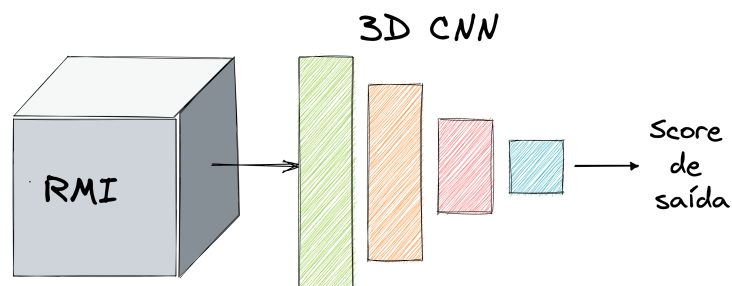


Figura 21 – Pipeline utilizada na abordagem *end-to-end* 3D.

### 4.2.3 Abordagem 2D (baseada em fatias)

Para a classificação ao nível de fatias, os testes serão realizados a partir de modelos baseados na arquitetura *ResNet* (assim como em (KORFIATIS *et al.*, 2017)) e na arquitetura *EfficientNet*. Diferentemente das abordagens anteriores, a análise do problema sob a ótica 2D faz com que o processo de classificação necessite de uma reformulação: a classificação deverá classificar cada fatia em três classes, sem tumor, com tumor MGMT metilado e com tumor e MGMT não metilado (necessitando do conhecimento da segmentação do tumor para correta construção).

Os dados são obtidos a partir da extração de “fatias” que compõe o volume de entrada. A partir dos dados utilizados, um volume de dimensões 240X240X155 gerará um conjunto de 155 imagens de dimensão 240X240. A última dimensão foi utilizada para “fatiamento” para possibilitar a obtenção de imagens de dimensões iguais. A Figura 22 apresenta como as fatias são organizadas em uma RMI.

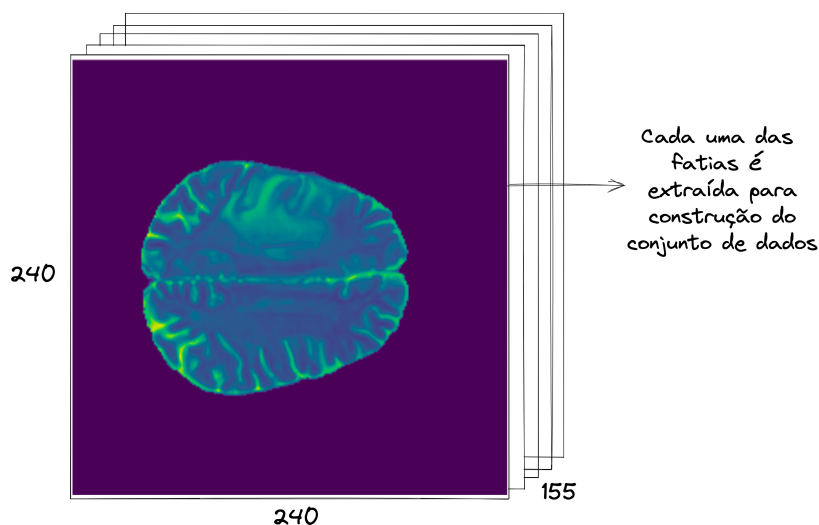


Figura 22 – Organização das fatias em uma RMI.

Como função custo, a entropia cruzada (4) será utilizada para definir os pesos da rede durante o treinamento. A camada de saída de todas as arquiteturas é definida como a função *softmax* (a qual gera saídas mutuamente exclusivas). No caso do problema em questão essa abordagem faz sentido, pois os estados possíveis de cada fatia são mutuamente exclusivos já que um paciente sem tumor não poderá ser diagnosticado com metilação no gene MGMT).

Como o treinamento é efetuado por fatia e a classificação deve ser feita ao nível de paciente, as previsões individuais precisam ser fundidas para geração de um *score* de saída. Para geração desse *score* um pós-processamento será aplicado. Essa etapa inicialmente irá remover as fatias classificadas como sem tumor (fatias com o maior *score* de saída referente a essa classificação) para depois gerar dois *scores* a partir de dois tipos de votação:

- *hard voting*: a geração do *score* de saída do paciente é feita com base na proporção de fatias classificadas como MGMT metilado em relação às classificadas como MGMT não metilado;
- *soft voting*: nessa abordagem a média dos *scores* para metilação e não metilação para cada paciente é obtida para posterior re-normalização (soma das médias deve ser 1).

A Figura 23 apresenta um fluxograma do treinamento e da geração das saídas ao nível de paciente.

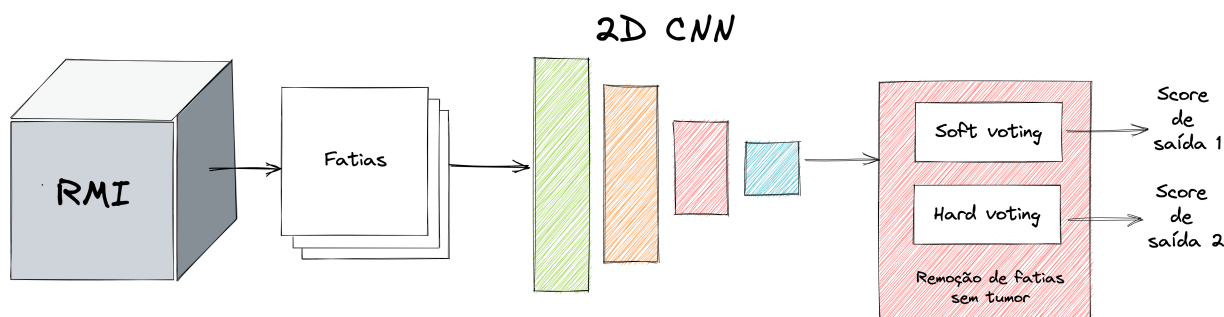


Figura 23 – Pipeline utilizada durante a abordagem baseada em fatias.

Todos os ajustes de hiperparâmetros e arquiteturas serão realizados com o intuito de se maximizar a métrica ROC-AUC, a nível de paciente, no conjunto de validação selecionado. Hiperparâmetros como taxa de aprendizado, regularização, tamanho de *batch*, etc. serão testados durante os experimentos. Caso seja necessário, *data augmentation* será utilizada nas imagens de entrada, construindo um conjunto de dados com maior variabilidade.

#### 4.2.4 Abordagem via segmentação semântica

Seguindo a linha apresentada no trabalho original (YOGANANDA *et al.*, 2021), uma arquitetura utilizada para segmentação semântica será utilizada para classificação individual de cada *voxel* pertencente ao tumor. Essa abordagem gera dois volumes de saída, um correspondendo ao gene não metilado e outro correspondendo ao gene metilado. A Figura 24 ilustra a abordagem utilizada pelos autores.

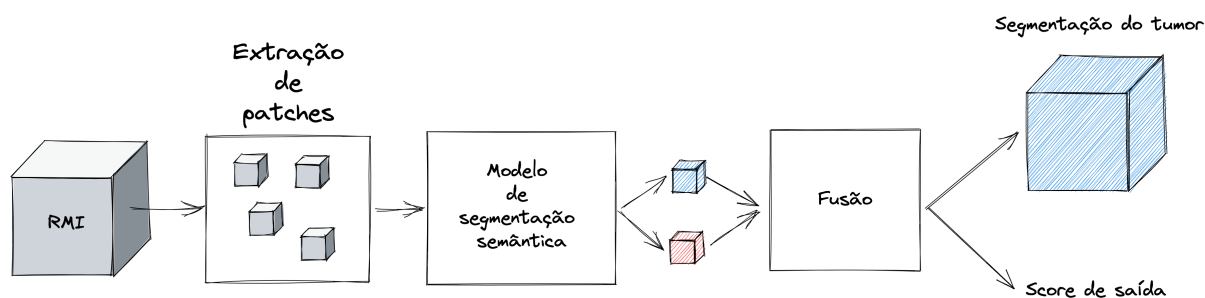


Figura 24 – Diagrama representando a abordagem de classificação via segmentação semântica.

No trabalho (YOGANANDA *et al.*, 2021) os autores utilizam uma arquitetura própria, pré-treinada na tarefa de classificação do *status* do gene IDH, chamada de *Dense U-Net* para treinamento na tarefa de classificação do *status* do gene MGMT. Essa arquitetura é baseada em uma conexão densa entre as camadas, a qual conecta cada uma das camadas convolucionais ao restante das camadas presentes, fazendo com que o fluxo de propagação do gradiente e de informação ao longo do treinamento sejam melhorados (HUANG *et al.*, 2017).

Devido à falta de informações no trabalho original sobre a arquitetura (como quantidade de filtros convolucionais em cada camada, tamanho do *kernel* utilizado em cada camada, etc.), as quais são imprescindíveis para à correta reprodução do modelo desenvolvido, os experimentos serão realizados com base em uma arquitetura U-Net (RONNEBERGER; FISCHER; BROX, 2015).

A arquitetura será submetida a variações na quantidade de filtros utilizada em cada camada, bem como a quantidade de etapas utilizadas nos blocos de *encoding* e *decoding*, visando a maximização da métrica ROC-AUC no conjunto de validação. A camada final será alterada para geração de dois volumes de saída.

Testes sem pré-treinamento e com pré-treinamento na tarefa de classificação do *status* do gene IDH serão realizados tanto no conjunto de dados BRATS 2021 quanto no conjunto de dados TCIA, visando a classificação do *status* do gene MGMT. Esses testes serão realizados com o intuito de se comparar os resultados obtidos em cada um dos conjuntos de dados.

Todos os treinamentos dessa abordagem serão baseados em *patches* de 32X32X32, com 75% de *overlap* e com no mínimo 1 *voxel* de tumor. Cada um desses *patches* será processado pelo modelo e um *score* será gerado (valor entre 0 e 1) após normalização por uma função sigmoide (3). A função custo utilizada é a entropia cruzada binária. Todos os

hiperparâmetros utilizados pelo modelo, bem como a necessidade de *data augmentation* serão ajustados conforme a necessidade, visando o melhor resultado de métrica ROC-AUC possível no conjunto de validação.

Os *patches* são extraídos em uma etapa *offline*, a qual gera todos os *patches* necessários para treinamento. Cada *patch* é extraído a partir de uma janela deslizante ao longo do volume de entrada. O deslizamento da janela sempre se inicia na posição  $(0, 0, 0)$  e vai varrendo todo o volume de entrada, de dimensões  $240 \times 240 \times 155$ . A Figura 25 exemplifica como os *patches* são gerados a partir de uma RMI. Devido ao *overlap*, o deslocamento em cada eixo necessário para varrer o volume deve ser feito com um passo de 8 *voxels*.

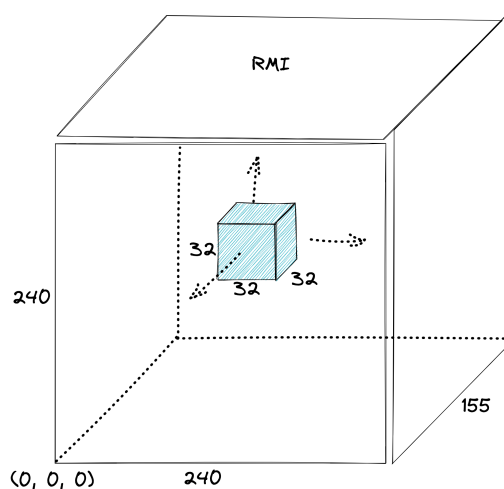


Figura 25 – Exemplificação da extração de *patches* a partir de uma RMI.

Para geração da máscara as previsões para cada um dos *patches* gerados serão comparadas *voxel a voxel* de forma se obter o maior *score* de saída para cada um dos *voxels*. Após isso, uma etapa de binarização será realizada: *scores* maiores que 0,5 serão transformados em 1 e *scores* menores em 0, resultando em uma máscara binária que pode ser utilizada para remoção de estruturas cerebrais fora da região do tumor.

Após as previsões, um pós-processamento será realizado com o intuito de fundir as previsões e gerar um *label* para o paciente (para posterior análise via métrica ROC-AUC). A fusão, assim como na abordagem anterior, será realizada com base em um esquema *soft voting* e também com base em um esquema *hard voting*. Devido à capacidade de geração da máscara do tumor, esses esquemas de votação serão realizados de duas maneiras: sem a utilização da máscara do tumor e com a utilização da máscara do tumor. A utilização da máscara permitirá a remoção de *voxels* fora do tumor para cômputo do *score* de saída de cada paciente. Os esquemas de votação *hard* e *soft* são definidos da seguinte forma:

- *hard voting* inicialmente somam-se todas as previsões para cada saída, e após isso se determina o *status* do *patch* a partir da comparação desses valores. Com isso, a geração do *score* de saída do paciente é feita com base na proporção de *patches*

classificados como MGMT metilado em relação às classificadas como MGMT não metilado;

- *soft voting*: nessa abordagem a soma (considerando todos os *patches*) dos *scores* para metilação e não metilação de cada paciente é obtida para uma posterior re-normalização (soma dos valores deve ser 1).

## CAPÍTULO 5

---

### Resultados e discussões

---

Nesta seção os resultados obtidos durante os experimentos serão apresentados bem como as discussões referentes a cada experimento.

O *hardware* utilizado para treinamento nos modelos baseados em redes neurais convolucionais foi uma GPU NVIDIA GeForce RTX 2080TI - 11GB. Para o treinamento de abordagens mais custosas computacionalmente uma GPU NVIDIA GeForce RTX 3080TI - 12GB foi utilizada. A utilização da GPU é necessária devido à sua capacidade de paralelização das operações, aumentando a velocidade do treinamento dos modelos.

Todos os processos de manipulação dos dados, construção e validação dos modelos foram realizados a partir de bibliotecas *python* de código livre, muito difundidas em projetos de *machine learning* e com desempenho comprovado ao longo de diversas soluções de sucesso.

O *framework* de *deep learning* escolhido para desenvolvimento foi o *PyTorch* (PASZKE *et al.*, 2019) devido à sua simplicidade de uso, difusão na comunidade científica e capacidade de construção e personalização de diferentes modelos.

Em todos os testes envolvendo *deep learning* os modelos utilizados foram baseados nas implementações desenvolvidas no *framework* de processamento de imagens médicas *monai* (MONAI CONSORTIUM, 2022). Esse projeto traz diferentes ferramentas úteis na área de processamento de imagens médicas e arquiteturas de *deep learning* já desenvolvidas.

Todas as transformações relacionadas à *data augmentation* na abordagem via segmentação semântica foram realizadas por meio da biblioteca de código aberto *Albumentations* (BUSLAEV *et al.*, 2020). Essa biblioteca implementa diferentes categorias de transformações e consegue lidar tanto com volumes quanto com problemas de segmentação (aplicando as transformações do volume de entrada na máscara também).



## 5.1 CLASSIFICAÇÃO A PARTIR DE *FEATURES* RADIÔMICAS

Nesta abordagem, o volume do tumor foi extraído e processado por uma *pipeline* de extração de *features*. Essas *features* são características que representam o tumor sob análise. A partir dessas características e do *status* MGMT de cada paciente, um modelo de classificação foi treinado. Essa abordagem necessita da segmentação do tumor para correta extração das características radiômicas do tumor.

Diferentes modelos com diferentes combinações de hiperparâmetros foram testados, com o intuito de se maximizar a métrica ROC-AUC no conjunto de validação evitando também o *overfitting*, o qual ocorria com devida facilidade devido à pequena quantidade de dados disponível.

O processamento do volume 3D do tumor pela biblioteca *PyRadiomics* gerou um total de 107 diferentes *features* para cada paciente. Todas essas características são valores reais não nulos que podem ser diretamente processados por modelos de aprendizado de máquina clássicos.

Três modelos de classificação foram testados: regressão logística, classificador baseado em *random forests* e classificador baseado em *gradient boosting*. Em ambos os casos a implementação utilizada foi a oriunda da biblioteca *sklearn* (PEDREGOSA *et al.*, 2011). Hiperparâmetros relacionados à regularização foram utilizados para evitar o *overfitting*, presente principalmente nos classificadores baseados em *random forests* e em *gradient boosting*.

Na regressão logística a constante de regularização foi alterada com o intuito de se reduzir o *overfitting*. Nos modelos de *random forests* e *gradient boosting* parâmetros como a profundidade, número de estimadores e total de amostras por nó foram alterados, com o intuito de se reduzir a complexidade dos modelos (e também reduzir o *overfitting*).

Outra técnica que se mostrou útil foi a remoção de *features* correlacionadas. Para cada grupo de características, com correlação maior que 80%, somente a *feature* de maior variância foi mantida. A partir dessa seleção o conjunto de *features* foi reduzido para 35. Essa abordagem melhorou o resultado de ambas abordagens, principalmente reduzindo o sobreajuste nos dados de treinamento.

Para treinamento do modelo de regressão logística a normalização apresentada na Equação (12) foi utilizada.

$$\hat{x} = \frac{x - \bar{x}}{\sigma} \quad (12)$$

Devido ao método de validação utilizado, os hiperparâmetros selecionados foram os que geraram o maior valor médio de ROC-AUC ao longo das simulações. Para cada conjunto de hiperparâmetros 100 simulações foram realizadas e as métricas no conjunto de treino e de validação a cada simulação salvas para análise posterior.

A Figura 26 apresenta os valores da métrica ROC-AUC obtidos ao longo das simulações para o modelo de regressão logística. Nota-se a distribuição mais ampla (com maior desvio-padrão) da métrica de validação em relação à métrica de treino. Em termos médios a ROC-AUC (%) obtida no treinamento foi de 68,62% e a ROC-AUC (%) obtida no conjunto de validação foi de 59,75%.

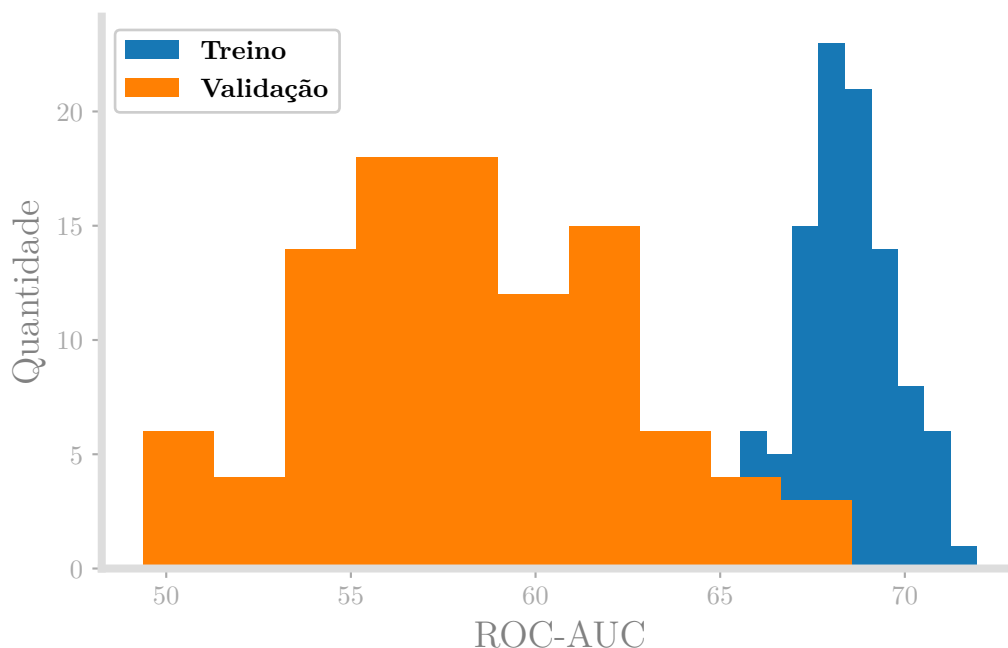


Figura 26 – Resultados da métrica ROC-AUC (%) ao longo das simulações para o modelo de regressão logística.

Os resultados apresentados na Figura 27 mostram o desempenho do modelo de classificação baseado em *gradient boosting*. Assim como na regressão logística, a variabilidade dos resultados no conjunto de validação também foi acentuada. Em termos de valores médios, no treinamento o valor obtido de ROC-AUC (%) foi de 72,94% e o valor obtido no conjunto de validação foi de 58,34%.

Na Figura 28 os resultados obtidos para o modelo de *random forests* podem ser observados. Dos três modelos testados esse é o que obteve os maiores valores durante o treinamento, mostrando certa facilidade para o *overfitting*. No conjunto de treinamento a métrica ROC-AUC (%) média foi de 75,00% e no conjunto de validação a seu valor foi de 57,86%.

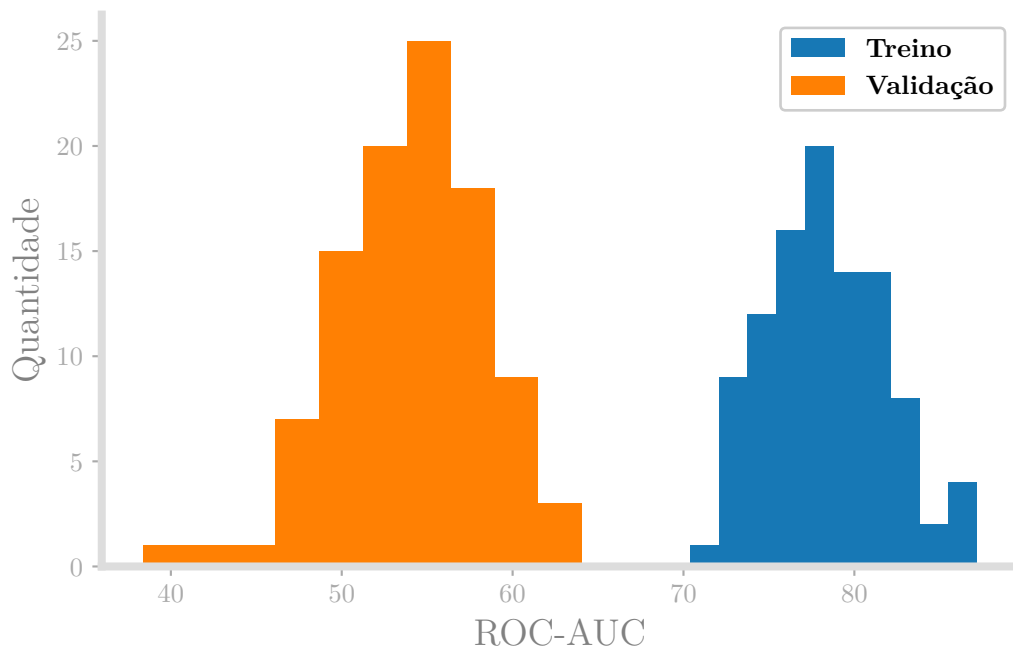


Figura 27 – Resultados da métrica ROC-AUC ao longo das 100 simulações para o modelo de *gradient boosting*.

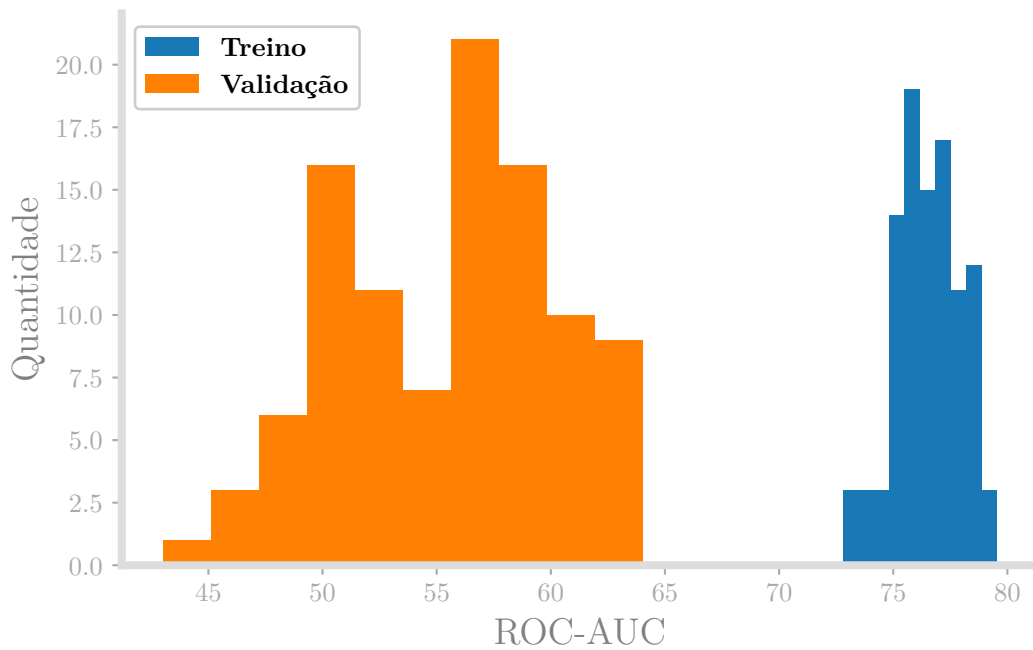


Figura 28 – Resultados da métrica ROC-AUC ao longo das simulações para o modelo de *random forests*.

A métrica no conjunto de teste também foi calculada com a mesma abordagem iterativa, gerando 100 conjuntos de previsões para o conjunto de teste. A Figura 29 apresenta a distribuição da ROC-AUC (%) no conjunto de testes para cada um dos modelos apresentados.

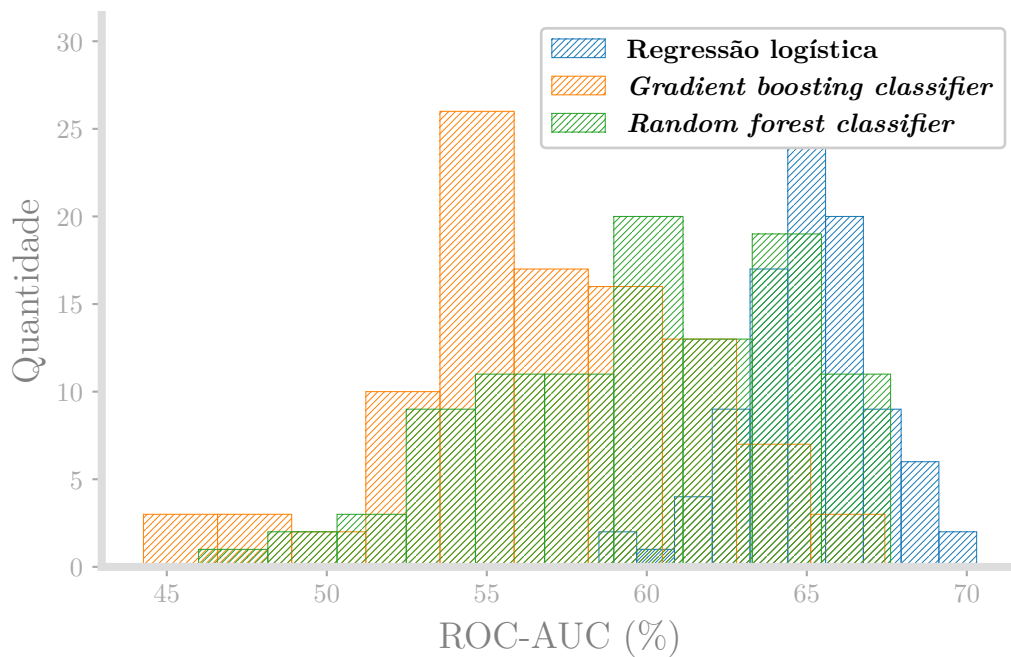


Figura 29 – Resultados da métrica ROC-AUC (%) no conjunto de testes ao longo das simulações com diferentes modelos.

A Figura 30 apresenta a comparação das curvas ROC, calculadas no conjunto de teste, para cada um dos classificadores. O modelo de regressão logística apresentou o melhor desempenho dentre os testados, tendo um desempenho superior ao dos demais em todos os pontos de operação.

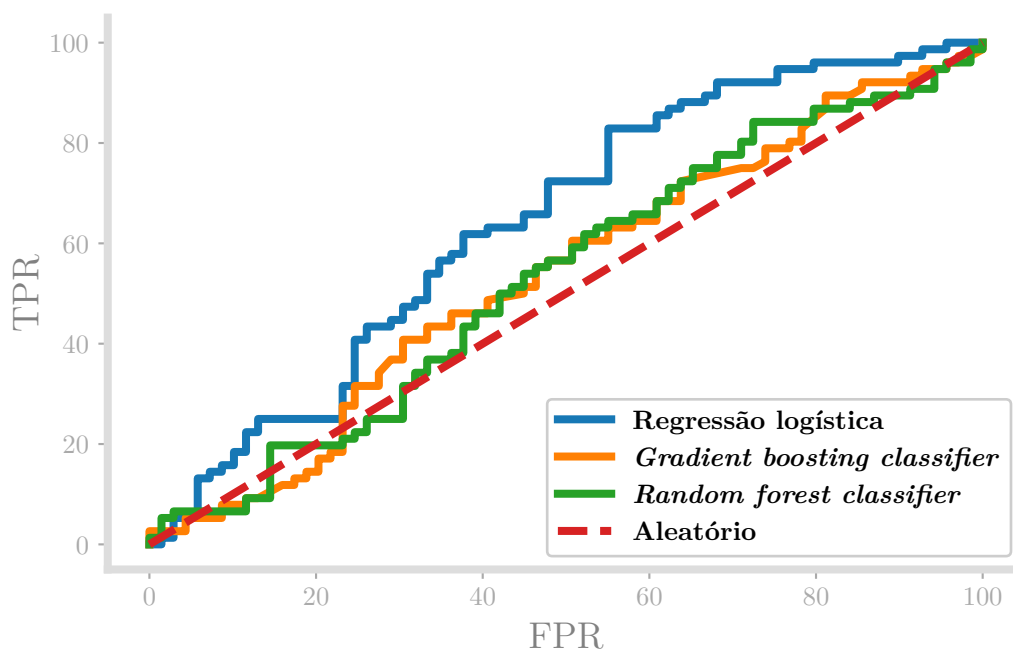


Figura 30 – Comparação da curva ROC para cada um dos modelos.

A tabela 1 sintetiza tanto o valor médio de cada métrica quanto o desvio-padrão obtido (representado pelo valor entre parênteses).

Tabela 1 – Resultados obtidos na abordagem baseada em extração de *features*.

	Treino	Validação	Teste
<b>Regressão logística</b>	68,42 (1,32)	58,17 (4,10)	65,02 (2,10)
<b>Gradient boosting</b>	78,24 (3,50)	54,08 (4,38)	56,93 (4,57)
<b>Random forests</b>	76,53 (1,38)	55,43 (4,70)	60,00 (4,69)

Nota-se a variabilidade nas métricas ao longo das simulações e nos diferentes modelos, bem como o valor elevado das métricas de teste em comparação com a métrica de validação, a qual está ligada aos níveis de variância elevados, característica do *overfitting*.

A regressão logística apresentou o melhor resultado no conjunto de teste além do menor *overfitting* e menor variabilidade nos resultados. Tanto o modelo baseado em *gradient boosting* quanto o modelo baseado em *random forests* obtiveram resultados semelhantes no conjunto de validação, com elevada variabilidade dos resultados.

## 5.2 CLASSIFICAÇÃO 3D END-TO-END

Essa abordagem visa classificar cada paciente individualmente, a partir de seu volume de entrada. Para essa abordagem os rótulos de cada paciente (MGMT metilado/MGMT não metilado) foram utilizados para treinamento. Diferentemente da solução anterior, essa

abordagem não necessita do conhecimento da segmentação do tumor, tornando sua aplicação mais direta.

Diferentes arquiteturas foram testadas, sendo uma arquitetura *ResNet* a selecionada. Essa arquitetura, chamada *ResNet10* é uma versão menor (de menor complexidade) das arquiteturas *ResNet*. A redução de complexidade do modelo foi necessária devido à facilidade de *overfitting* encontrada durante o treinamento de modelos com maior número de parâmetros, além de instabilidade (variação muito grande) na métrica de validação ao longo do treinamento.

Para treinamento todos os volumes foram normalizados a partir da Equação (12). Essa normalização garante um treinamento mais suave e uma melhor atualização dos pesos do modelo a cada iteração.

Diferentes hiperparâmetros foram testados com o intuito de se maximizar a métrica no conjunto de validação e de se obter um treinamento estável ao longo das épocas (com evolução suave da função custo), dentre esses hiperparâmetros pode-se citar: taxa de aprendizado, número de épocas, constante de regularização e tamanho do *batch*.

O aumento do tamanho do *batch* proporcionou uma maior estabilidade no treinamento, mostrando a tendência de aumento na métrica de validação ao longo das épocas. Esse aumento provavelmente ocorreu devido à melhor estimativa dos parâmetros utilizados pela camada de *batch normalization* utilizada pela arquitetura.

Diferentes tamanhos de *batch* foram testados, partindo de um tamanho de 1 volume por *batch* e chegando a 64 volumes. À medida que valores maiores de *batch* eram testados maior o consumo da memória GPU durante o treinamento. A partir de um tamanho de *batch* de 14 os volumes de entrada foram redimensionados de 128X128X128 para 64X64X64, para o treinamento não extrapolar a capacidade de memória da GPU.

Mesmo com a alteração de diferentes hiperparâmetros (e diminuição da variação das métricas) e com a redução de complexidade do modelo o *overfitting* continuava presente durante o treinamento. Com o intuito de se aumentar a variabilidade do conjunto de treinamento, diferentes técnicas de *data augmentation* foram testadas, como inserção de ruído aleatório, rotações aleatórias, etc (sempre aplicadas no volume completo de entrada). Com os testes observou-se que a utilização de um *zoom* aplicado aleatoriamente no volume de entrada conseguia não só reduzir significativamente o *overfitting* como suavizar a evolução das métricas ao longo do treinamento.

Essa técnica de *data augmentation* pode ser observada na Figura 31. O *zoom* é aplicado aleatoriamente (com 50% de chance de ser aplicado a cada volume) tanto como um *zoom out* quanto como um *zoom in*, de forma independente em cada eixo.

A Tabela 2 apresenta o conjunto de hiperparâmetros selecionados nesse experimento.

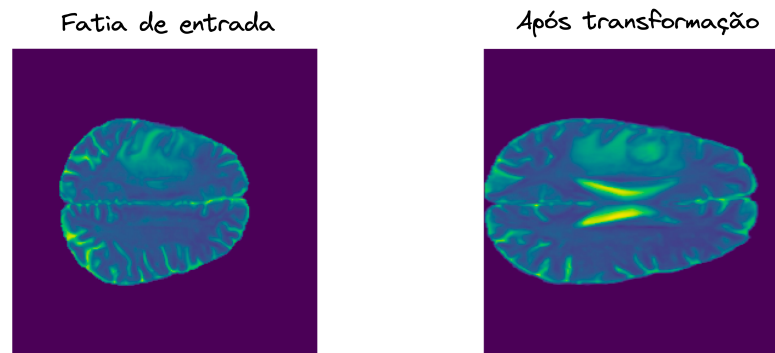


Figura 31 – Resultados da aplicação da técnica de *data augmentation* aplicada durante o treinamento.

Tabela 2 – Hiperparâmetros obtidos na abordagem *end-to-end*.

<b>Arquitetura utiliza</b>	ResNet10
<b>Taxa de aprendizado</b>	0,001
<b>Número de épocas</b>	30
<b>Tamanho do <i>batch</i></b>	32
<b>Otimizador</b>	<i>Adam</i>
<b>Função custo</b>	Entropia cruzada binária
<b><i>Data augmentation</i></b>	<i>Zoom aleatório</i>

A Figura 32 apresenta a evolução da função custo ao longo das épocas. Nota-se que o custo de validação atingiu seu valor mínimo logo após as primeiras épocas enquanto o custo de treino continuou sendo reduzido durante todo o período de treinamento.

A métrica ROC-AUC durante o treinamento pode ser observada na Figura 33. Pode-se perceber a tendência de aumento da métrica ao longo das épocas. O melhor valor dessa métrica no conjunto de validação ocorreu na época 13, representada pela linha tracejada nas figuras. Após isso, a métrica de validação atinge um platô e o custo começa a aumentar suavemente.

Pode-se notar a pequena variação no custo ao longo das épocas, a qual já representou ganhos em termos de métrica ROC-AUC. Isso ocorre devido à natureza do cálculo dessa métrica: enquanto a função custo penaliza o valor numérico das predições, a métrica ROC-AUC apenas mede se a ordenação das predições está correta (i.e., se *scores* de pacientes com metilação estão maiores que *scores* de pacientes sem metilação).

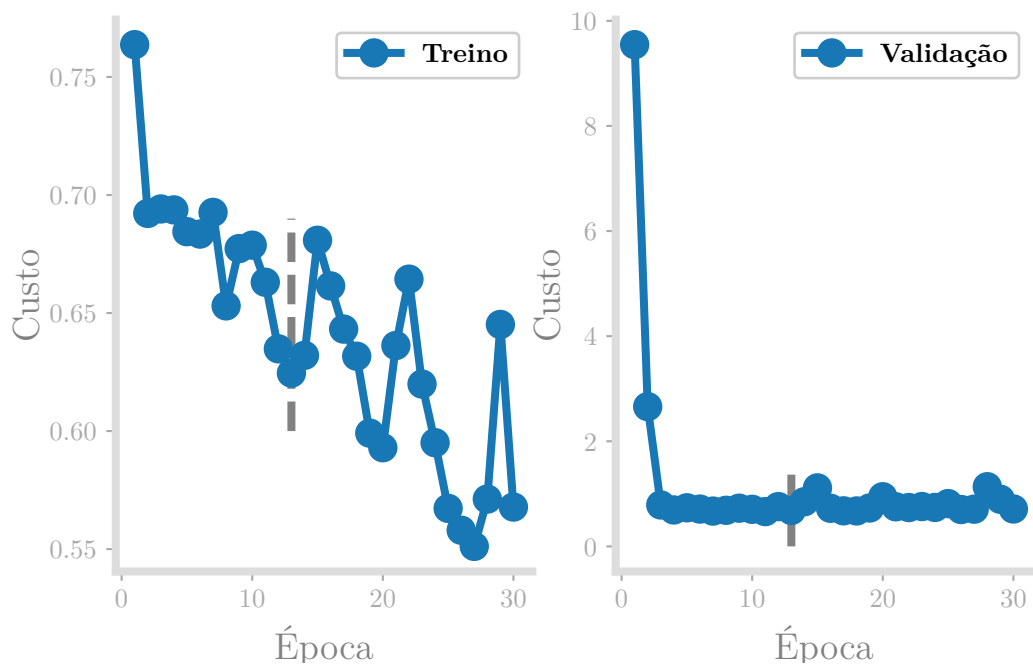


Figura 32 – Custo ao longo das épocas nos conjuntos de treinamento e validação para a abordagem *end-to-end* 3D.

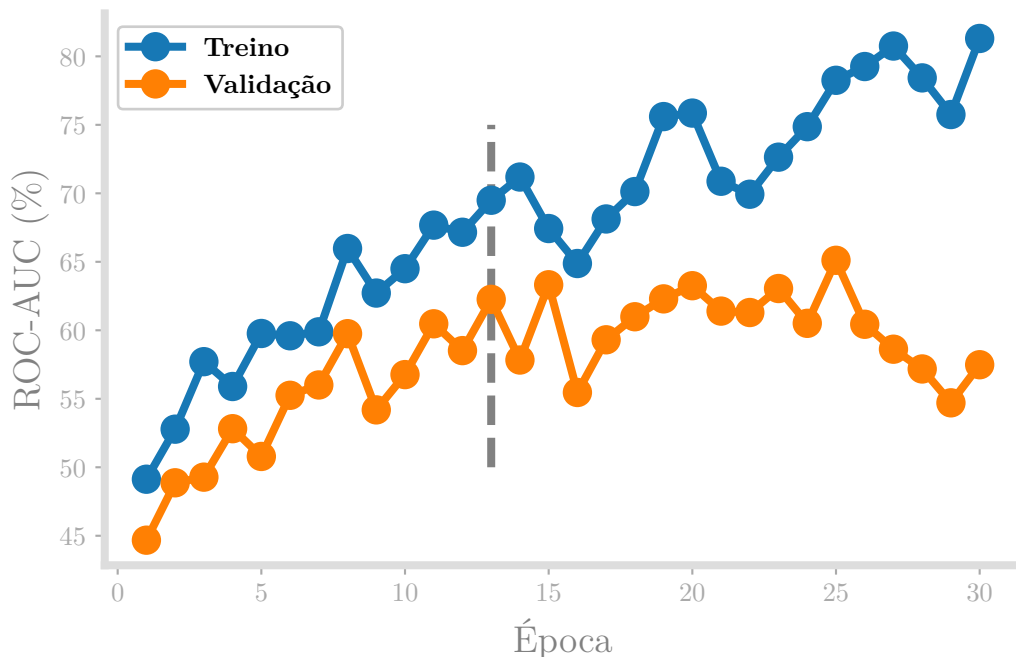


Figura 33 – Métrica ROC-AUC (%) ao longo das épocas nos conjuntos de treinamento e validação para a abordagem *end-to-end* 3D.

A Figura 34 apresenta as curvas ROC obtidas para o conjunto de teste, na época 13.



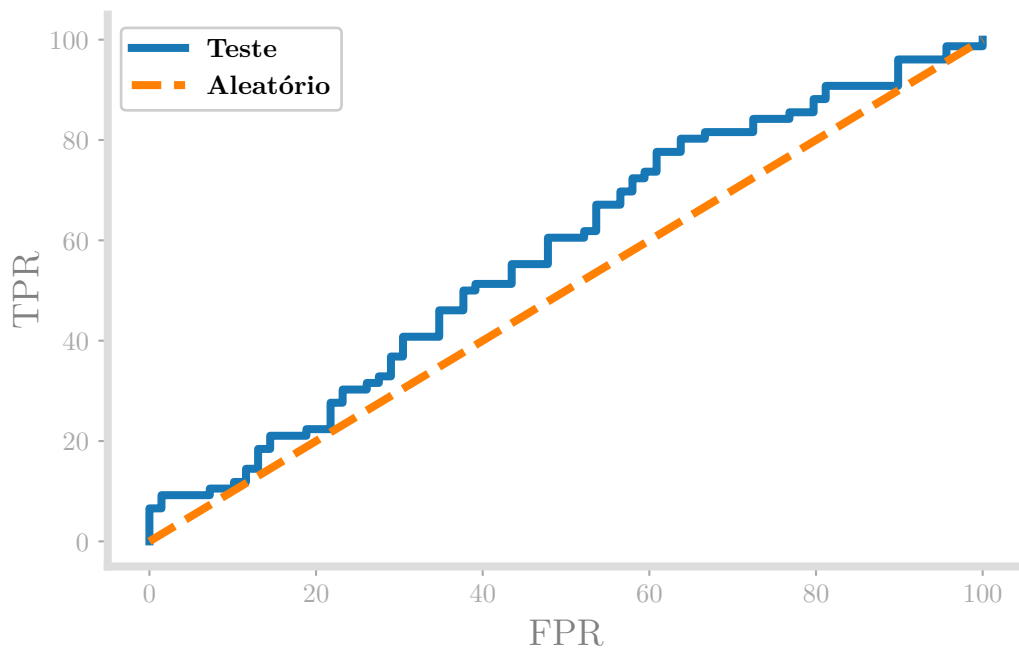


Figura 34 – Curvas ROC para os conjuntos de treino, validação e teste.

A Tabela 3 apresenta os resultados em termos da métrica ROC-AUC (%) para essa abordagem. Nota-se a proximidade entre os resultados obtidos no conjunto de validação e no conjunto de teste.

Tabela 3 – Resultados em termos de métrica ROC-AUC (%) obtidos na abordagem *end-to-end* 3D.

	Treino	Validação	Teste
<i>End-to-end</i>	71,60	62,26	58,46

### 5.3 CLASSIFICAÇÃO 2D BASEADA EM FATIAS

A terceira abordagem testada visa classificar cada uma das fatias que compõe o volume de entrada (ressonância magnética do paciente). A partir dessa classificação individual o *status* do paciente pode ser inferido através de uma etapa de pós-processamento.

A classificação é realizada para determinar se a fatia não tem tumor ou, caso tiver um tumor, classificá-lo em MGMT metilado ou MGMT não metilado, tornando a abordagem um problema de classificação multiclasse. As fatias de entrada são oriundas de um volume de tamanho 240x240x155 o qual gera fatias de dimensão 240X240 *pixels*.

Antes do treinamento uma etapa de seleção de fatias foi realizada. Para uma fatia ser incluída no treinamento (e na inferência) ela deve ter ao menos 5028 *pixels* não nulos (70% das fatias presentes no conjunto de treinamento satisfazem essa condição). Isso foi útil para reduzir a quantidade de ruído inserido por fatias com pouca ou nenhuma área cerebral presente.

Mesmo com a remoção de fatias com pouca ou nenhuma área cerebral o processo de treinamento não apresentava sinais de aprendizado ao longo das épocas. Para auxiliar ainda mais o modelo, informações da máscara do tumor foram utilizadas para filtrar as fatias com poucos *pixels* de tumor. Fatias com menos de 50 *pixels* de tumor (em média a quantidade de *pixels* com tumor é de 683) são removidas durante a etapa de treinamento e durante a etapa de inferência. Caso a fatia seja sem tumor, ela é incluída sem passar por esse filtro.

O conjunto de treino gerado após extração e seleção de fatias é composto por 33 970 fatias, sendo 43% sem tumor, 30% com MGMT metilado e 27% com MGMT não metilado. O conjunto de teste é composto por 15 104 fatias com a mesma distribuição de classes.

Assim como na abordagem 3D, arquiteturas baseadas em *ResNet* e *EfficientNet* para classificação foram testadas, além de combinações de hiperparâmetros e de diferentes técnicas de treinamento. As arquiteturas do tipo *ResNet* obtiveram os melhores resultados no conjunto de validação, além de uma melhor estabilidade das métricas ao longo das épocas.

O modelo selecionado foi uma *ResNet34*. Totalmente treinada no conjunto de dados composto por fatias de RMI. Modelos baseados na arquitetura *EfficientNet* (*EFNET-B0*, *EFNET-B3* e *EFNET-B7*) não obtiveram resultados interessantes durante nenhum experimento.

Para treinamento, as imagens foram normalizadas com base na Equação (12), assim como nas abordagens anteriores. O tamanho do *batch* utilizado para treinamento foi de 32 fatias. Valores maiores degradaram o desempenho obtido. A função custo utilizada foi a entropia cruzada, a qual recebe como entrada a saída de uma função *softmax*.

Três técnicas de *data augmentation* foram combinadas. Assim como no caso anterior, o uso de *data augmentation* reduziu o *overfitting* (que surgia principalmente com taxas de aprendizado menor que 0,001) e a variação das métricas ao longo do treinamento. As transformações utilizadas foram *zoom* aleatório — apresentado na Figura 31, rotações aleatórias (Figura 35) e ruído *gaussiano* (Figura 36).

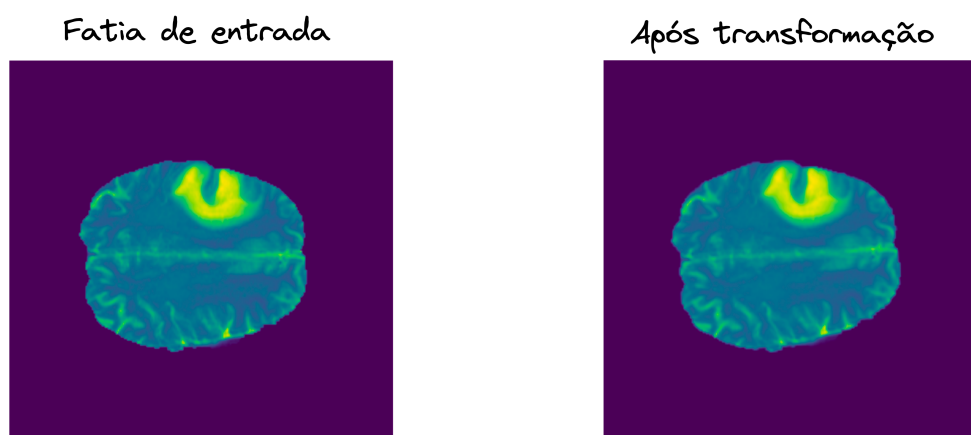


Figura 35 – Resultados de uma rotação aleatória aplicada na imagem original.

O grau de rotação é amostrado uniformemente do intervalo -10 a 10 graus. As rotações

podem ser aplicadas tanto ao eixo horizontal quanto ao vertical, com uma probabilidade de aplicação de 50%.

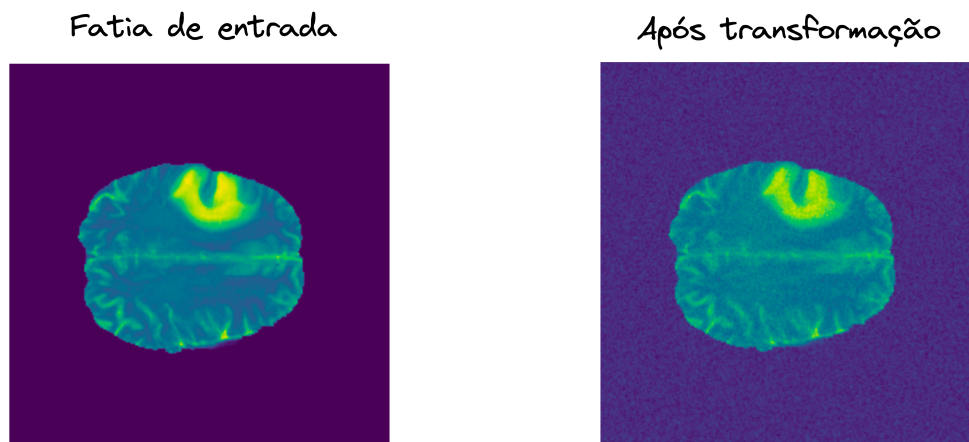


Figura 36 – Resultados da adição de um ruído gaussiano na imagem original.

Além de *data augmentation*, regularização L2 teve de ser inserida ao longo do treinamento. Essa regularização adiciona um termo dependente dos pesos do modelo na função custo, aumentando a capacidade de generalização do modelo treinado. Um parâmetro  $\lambda$  pondera o peso dessa regularização. A Equação (13) apresenta como a regularização utilizada é combinada com a função custo original,  $\mathcal{L}(\theta)$  representa a função custo original e  $\omega_i$  representa cada um dos pesos do modelo.

$$\hat{\mathcal{L}}(\theta) = \mathcal{L}(\theta) + \lambda \sum_{i=1}^P \omega_i^2 \quad (13)$$

Os hiperparâmetros selecionados nesse experimento estão apresentados na Tabela 4.

Tabela 4 – Hiperparâmetros obtidos na abordagem baseada em classificação de fatias.

<b>Arquitetura utiliza</b>	ResNet34
<b>Taxa de aprendizado</b>	0,001
<b>Número de épocas</b>	15
<b>Regularização L2</b>	0,0001
<b>Tamanho do <i>batch</i></b>	16
<b>Otimizador</b>	<i>Adam</i>
<b>Função custo</b>	Entropia cruzada
<b><i>Data augmentation</i> 1</b>	<i>Zoom</i> aleatório
<b><i>Data augmentation</i> 2</b>	Rotações aleatórias
<b><i>Data augmentation</i> 3</b>	Adição de ruído gaussiano

A Figura 37 apresenta o custo a cada época. Nota-se que o custo de validação se reduz, com oscilações, ao longo das épocas e começa a aumentar a partir da época 14 (representada pela linha tracejada).

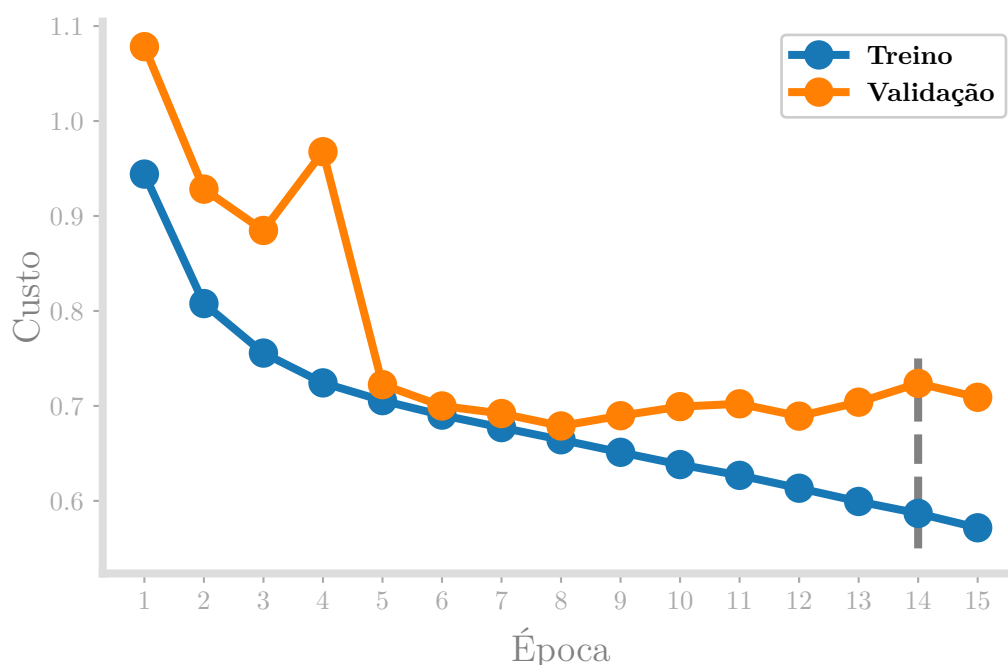


Figura 37 – Evolução da função custo ao longo das épocas nos conjuntos de treinamento e validação para a abordagem baseada em fatias.

A Figura 38 apresenta a métrica ROC-AUC ao longo das épocas. Tanto a evolução obtida através de uma votação *soft* como a evolução por uma votação *hard* tiveram uma tendência de crescimento ao longo das épocas até estabilização da métrica de validação por volta de um valor de 60%. Um treinamento com mais do que 14 épocas acaba fazendo com que o custo de validação aumente, degradando o desempenho no conjunto de validação.

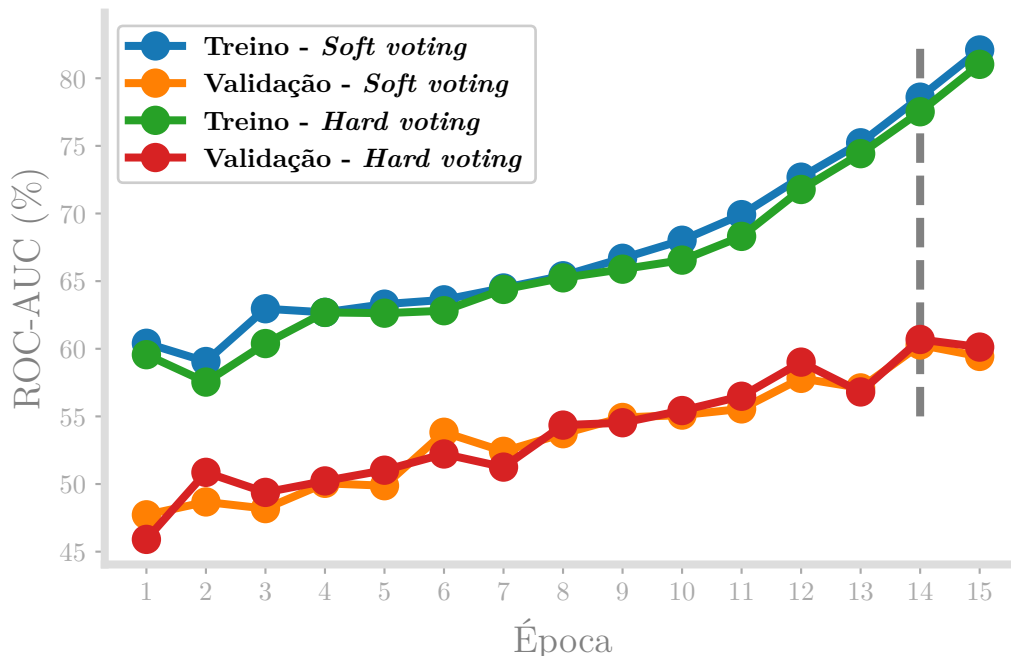


Figura 38 – Métrica ROC-AUC (%) ao longo das épocas nos conjuntos de treinamento e validação para a abordagem baseada em fatias.

A Tabela 5 sintetiza os resultados obtidos na época 14 para essa abordagem. Nota-se a proximidade entre as métricas calculadas com diferentes formas de votação.

Tabela 5 – Resultados em termos de métrica ROC-AUC (%) obtidos na abordagem baseadas em fatias.

	Treino	Validação
<b>Votação <i>soft</i></b>	78,60	60,28
<b>Votação <i>hard</i></b>	77,52	60,68

A técnica de votação *hard* conseguiu obter a melhor métrica no conjunto de validação. A Figura 39 apresenta a curva ROC obtida para o conjunto de teste, com uma métrica ROC-AUC de 60,55%

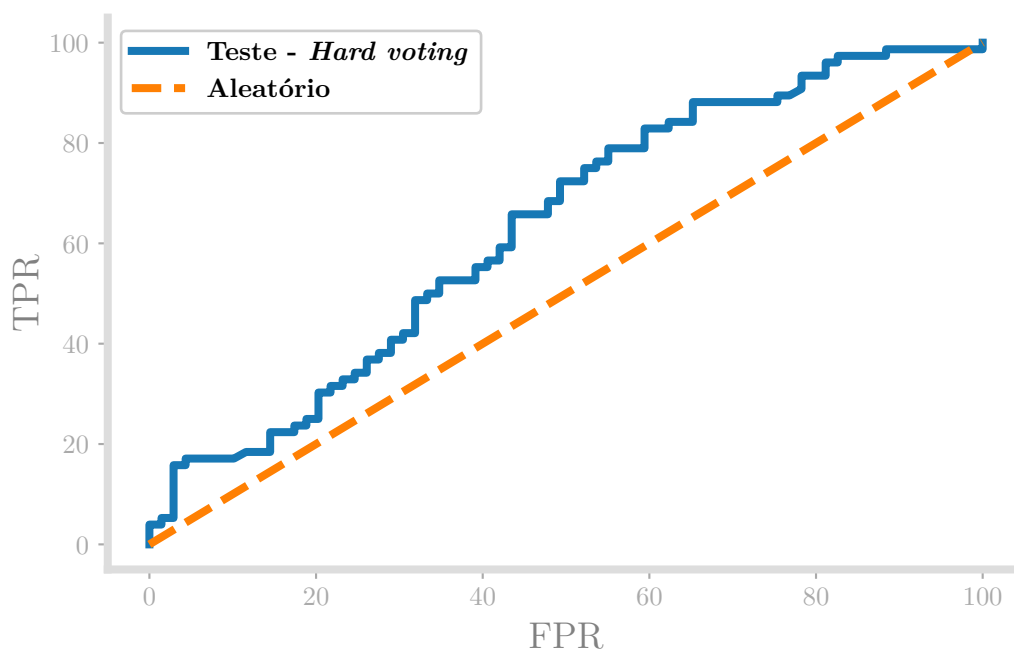


Figura 39 – Curvas ROC o experimento de classificação 2D baseada em fatias.

#### 5.4 CLASSIFICAÇÃO A PARTIR DE SEGMENTAÇÃO SEMÂNTICA

A quarta e última abordagem utiliza um modelo de segmentação semântica para geração de máscaras para o tumor presente no paciente. A partir das segmentações geradas, a classificação do paciente pode ser determinada a partir de uma etapa de votação, a qual gera um *score* de saída para cada paciente.

Inicialmente serão apresentados os resultados da classificação do *status* do gene MGMT no conjunto de dados BRATS 2021 e sem a utilização de pré-treinamento na tarefa de classificação do *status* do gene IDH. Após isso, os resultados oriundos da utilização de um pré-treino serão apresentados. O último resultado será referente à classificação do gene MGMT no conjunto de dados TCIA, assim como feito em (YOGANANDA *et al.*, 2021).

##### 5.4.1 Sem pré-treinamento

Para treinamento da abordagem baseada em segmentação semântica, *patches* de tamanho 32X32X32, com 24 *voxels* de *overlap* foram extraídos da RMI original. Tanto *patches* do cérebro quanto *patches* da máscara foram extraídos. Assim como em (YOGANANDA *et al.*, 2021) uma etapa de seleção, baseada na quantidade de *voxels* do tumor foi utilizada para reduzir o ruído no conjunto de dados. Os *patches* utilizados para treinamento devem ter ao menos 1 *voxel* de tumor, devido a isso a quantidade de *patches* varia conforme o paciente selecionado.

O conjunto de dados utilizado para treinamento foi constituído por 538 185 *patches*.

Desses, 52% estão com *status* positivo para a metilação do gene MGMT enquanto que 48% estão com *status* negativo. O conjunto de teste é composto por 177 822 *patches* com 54% deles pertencentes à classe MGMT metilado.

O propósito do modelo de segmentação semântica foi a geração de duas máscaras. Uma delas é gerada quando o *status* MGMT for metilado enquanto a outra é gerada quando o *status* for não metilado. Após a geração dessas máscaras uma etapa de votação, baseada nos *scores* de cada máscara, foi realizada para determinação da classificação do paciente.

A arquitetura selecionada para segmentação semântica é baseada na arquitetura U-Net, a qual apresentou as melhores métricas durante os experimentos preliminares. Os experimentos visaram a maximização da métrica ROC-AUC no conjunto de validação ao nível de paciente. Para se atingir esse objetivo, diferentes hiperparâmetros foram ajustados além de técnicas de *data augmentation* (quando necessárias). Variações na arquitetura de segmentação semântica, como profundidade e funções de ativação, também foram testadas para maximização da métrica.

Para geração da classificação ao nível de paciente, uma votação majoritária baseada nos *scores* de cada *patch* gerado foi realizada. A geração desse *score* é baseada em *soft* e *hard voting*, tanto utilizando a máscara do tumor gerado quanto não utilizando.

Logo nos primeiros testes a necessidade de *data augmentation* se mostrou necessária devido à facilidade de *overfitting* em menos do que 5 épocas de treinamento. Além das técnicas utilizadas anteriormente, outras foram utilizadas para aumentar a variabilidade do conjunto de dados: rotação e translações aleatórias, *blur*, e *flip* horizontal e vertical. A Figura 40 apresenta uma fatia de 128X128 de um *patch* de 128X128X128 após aplicação de algumas dessas transformações. Essas transformações também são aplicadas às segmentações utilizadas para garantir o *match* entre os dados utilizados para treino.

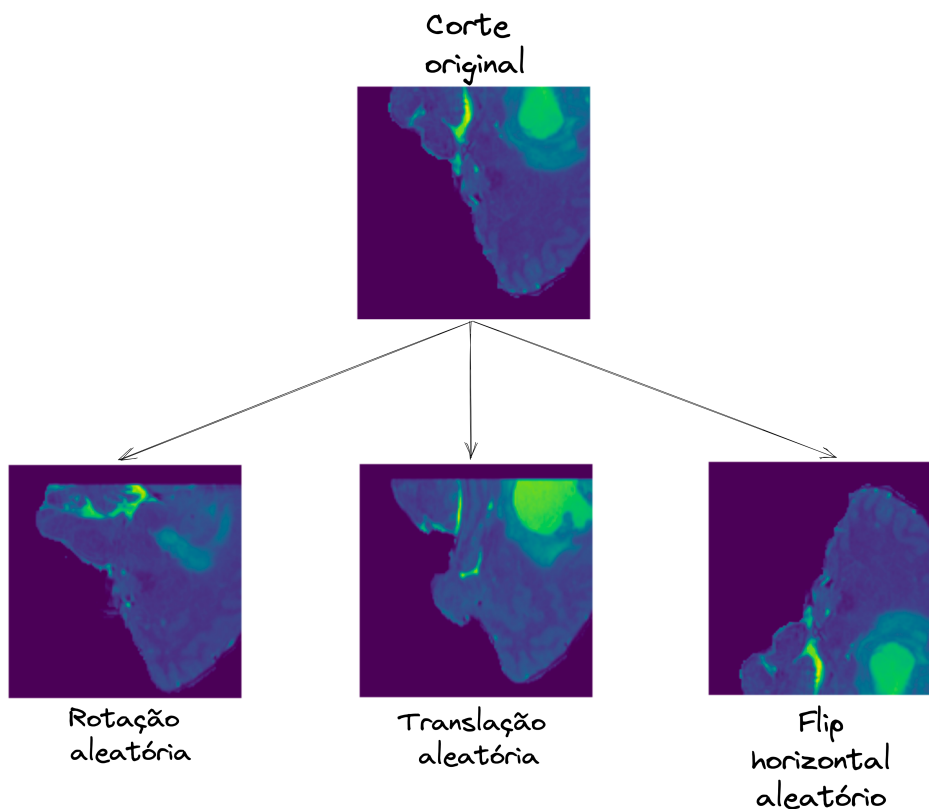


Figura 40 – Resultado das transformações de *data augmentation* utilizadas na abordagem baseada em segmentação semântica.

O tamanho de *batch* selecionado foi de 256 *patches*, com uma taxa de aprendizado de 0,0001. Valores maiores de taxa de aprendizado facilitavam ainda mais a ocorrência de *overfitting* durante o treinamento. Diferentemente das abordagens anteriores, devido ao elevado tamanho de *batch* a evolução da função custo foi mais suave ao longo das épocas, independentemente do conjunto de hiperparâmetros selecionados. Todos os *patches* foram normalizados a partir da Equação (12).

Os hiperparâmetros selecionados nesse experimento, bem como as etapas de *data augmentation* e a arquitetura utilizada podem ser observados na Tabela 6.

Tabela 6 – Hiperparâmetros obtidos na abordagem baseada em segmentação semântica.

Arquitetura utiliza	U-Net
Taxa de aprendizado	0,0001
Número de épocas	30
Tamanho do <i>batch</i>	256
Otimizador	<i>Adam</i>
Função custo	Entropia cruzada
<i>Data augmentation</i> 1	<i>Zoom</i> aleatório
<i>Data augmentation</i> 2	Rotações aleatórias
<i>Data augmentation</i> 3	Adição de ruído gaussiano
<i>Data augmentation</i> 4	Translação aleatória
<i>Data augmentation</i> 5	Adição de <i>blur</i>
<i>Data augmentation</i> 6	<i>Flip</i> horizontal
<i>Data augmentation</i> 7	<i>Flip</i> vertical



A evolução da função custo, e as métricas de treino e validação com todas as técnicas de votação podem ser observadas nas Figuras 41, 42 e 43, respectivamente.

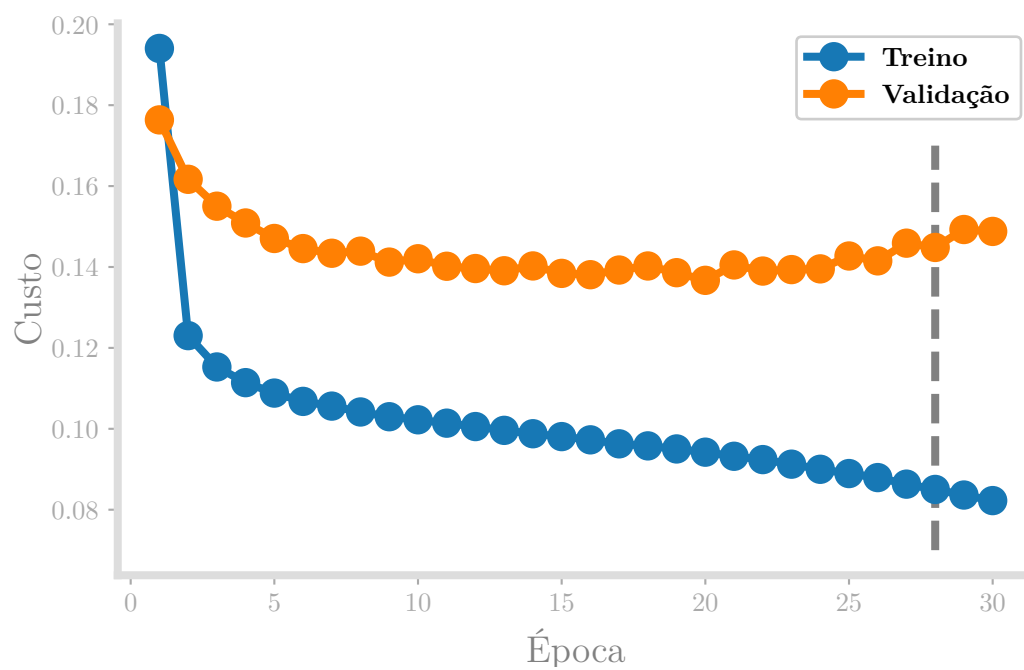


Figura 41 – Evolução da função custo ao longo das épocas nos conjuntos de treinamento e validação para a abordagem baseada em segmentação semântica sem pré-treinamento.

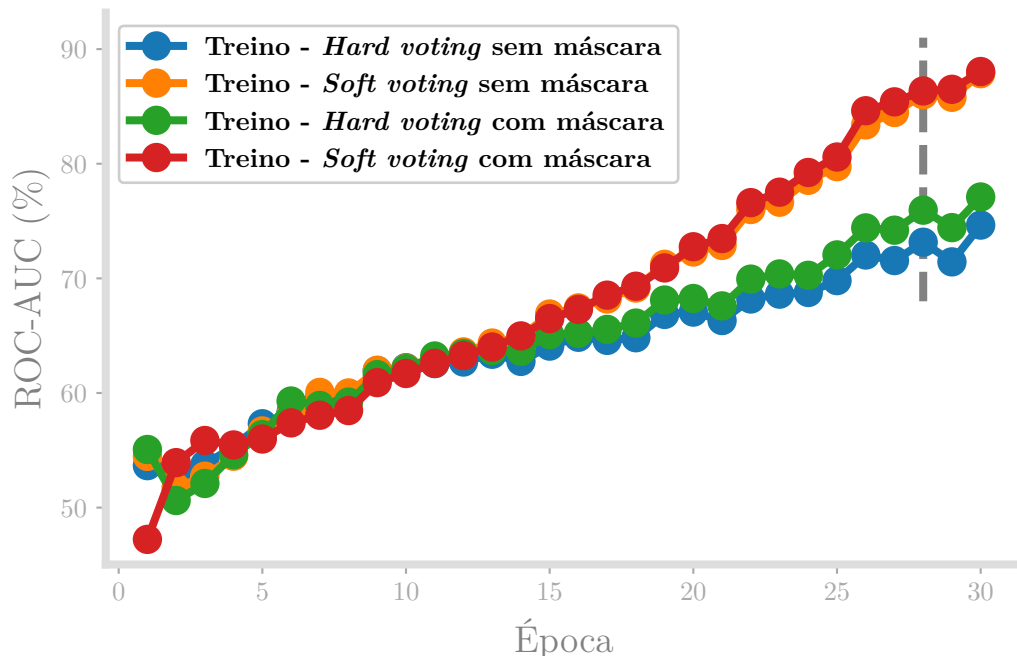


Figura 42 – Evolução da métrica ROC-AUC (%) ao longo das épocas no conjunto de treinamento para a abordagem baseada em segmentação semântica sem pré-treinamento.

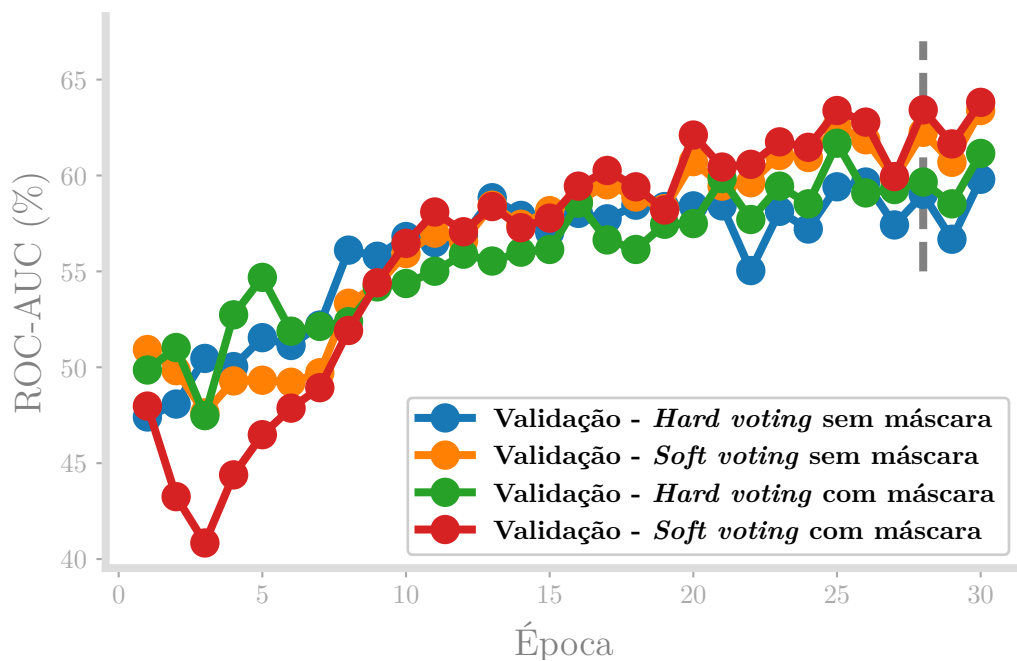


Figura 43 – Evolução da métrica ROC-AUC (%) ao longo das épocas no conjunto de validação para a abordagem baseada em segmentação semântica sem pré-treinamento.

Em 8 épocas o custo de validação se estabiliza, porém, o treinamento prossegue devido

ao aumento da métrica ROC-AUC no conjunto de validação. O treinamento foi interrompido na época 30 devido ao significativo aumento do custo de validação. Durante todo o treinamento as métricas evoluem de maneira semelhante, sendo que, nas épocas finais, as métricas baseadas em *soft voting* acabaram resultando em melhores resultados.

Os resultados obtidos, referentes a época 28, podem ser observados na Tabela 7.

Tabela 7 – Resultados em termos de ROC-AUC (%) obtidos na abordagem baseadas em segmentação semântica.

	Treino	Validação
<i>Soft voting</i> sem máscara	86,01	62,25
<i>Hard voting</i> sem máscara	73,15	59,01
<i>Soft voting</i> com máscara	86,35	63,42
<i>Hard voting</i> com máscara	75,95	59,65

O uso de *soft voting*, com a máscara do tumor, conseguiu obter melhores resultados no conjunto de validação. A Figura 44 apresenta a curva ROC obtida para essa abordagem, no conjunto de teste, que obteve uma métrica ROC-AUC de 56,44%.

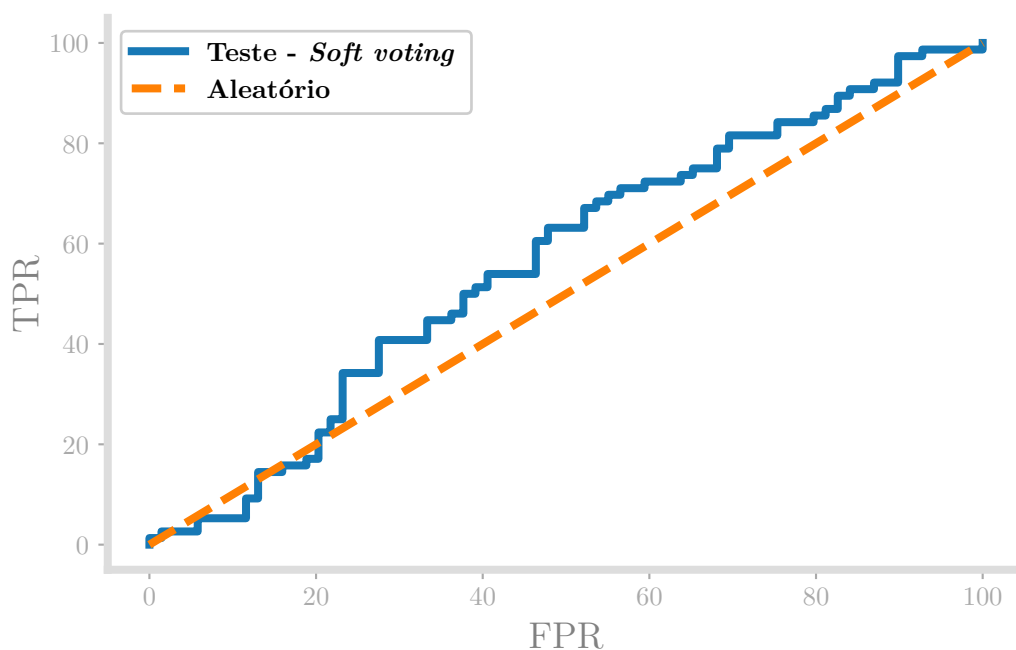


Figura 44 – Curvas ROC para a abordagem baseada em segmentação semântica.

#### 5.4.2 Com pré-treinamento

Para identificar o efeito do pré-treino nas métricas de classificação do *status* do gene MGMT essa mesma arquitetura foi pré-treinada no conjunto de dados TCIA, para classificar o *status* do gene IDH. Nesse treinamento a arquitetura U-Net foi mantida, sendo que a taxa de aprendizado foi alterada para 0,001. Os demais hiperparâmetros foram mantidos (bem como

*data augmentation* utilizada). Na tarefa de classificação do *status* do gene IDH os seguintes resultados de ROC-AUC (%) foram obtidos: 64,96% no conjunto de dados de treino e 65,62% no conjunto de dados de teste, todas as métricas medidas com base em *hard voting* sem a utilização da máscara do tumor.

A partir do modelo pré-treinado um novo treinamento foi realizado por 30 épocas no conjunto de dados BRATS 2021, na tarefa de classificação do *status* do gene MGMT. A função custo ao longo das épocas e as métricas de treino e validação, com todas as técnicas de votação, podem ser observadas nas Figuras 45, 46 e 47, respectivamente. A linha tracejada, novamente na época 28, representa a época utilizada para análise dos resultados.

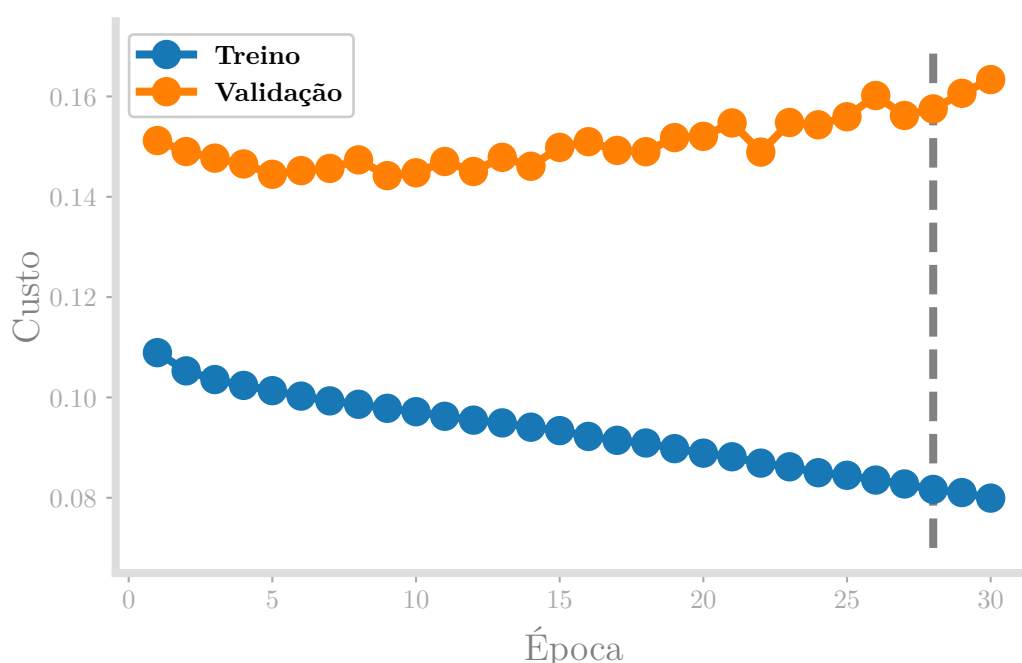


Figura 45 – Evolução da função custo ao longo das épocas nos conjuntos de treinamento e validação para a abordagem baseada em segmentação semântica com pré-treinamento.

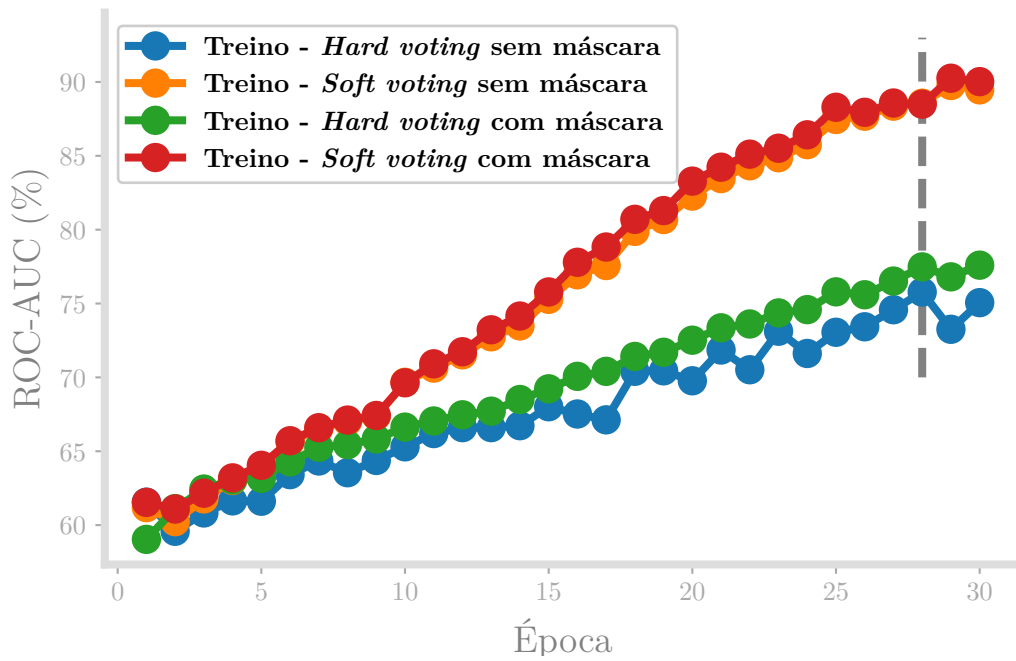


Figura 46 – Evolução da métrica ROC-AUC ao longo das épocas no conjunto de treinamento para a abordagem baseada em segmentação semântica com pré-treinamento.

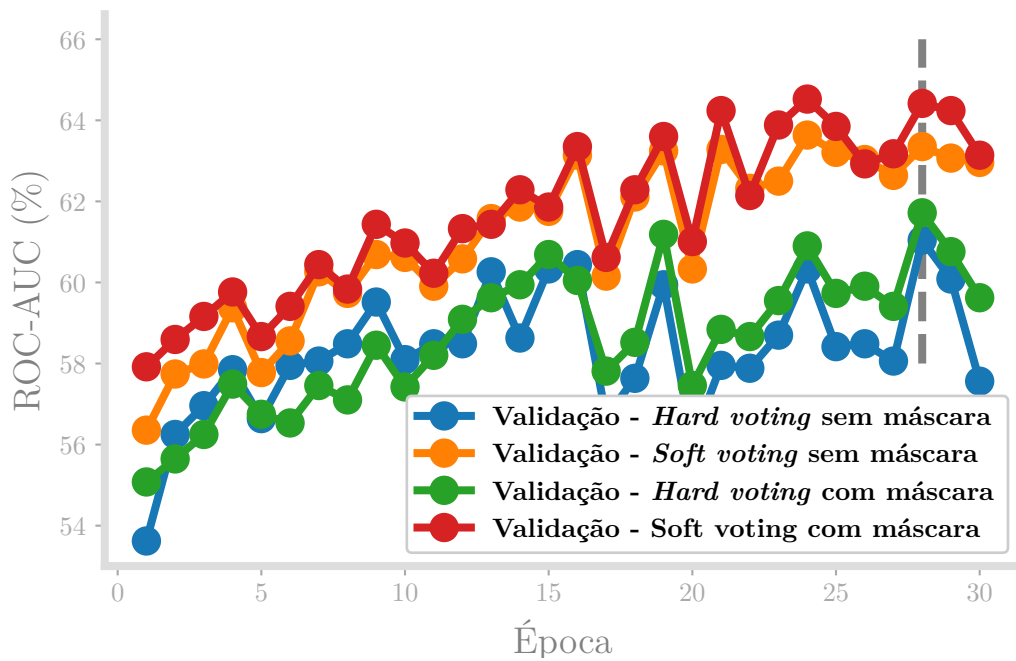


Figura 47 – Evolução da métrica ROC-AUC ao longo das épocas no conjunto de validação para a abordagem baseada em segmentação semântica com pré-treinamento.

Nota-se que tanto o custo de treinamento quanto o custo de validação iniciaram em

valores mais baixos do que no experimento com um modelo sem pré-treino (Figura 41). A partir da época 12, o custo de validação começa a aumentar suavemente sem atrapalhar a evolução das métricas de validação, conforme pode-se observar na Figura 47. As métricas baseadas em votação *soft* obtiveram valores de cerca de 90% no final do treinamento e de cerca de 60% no conjunto de validação. A métrica final de validação atingiu um platô na mesma região que a abordagem sem pré-treinamento.

A Tabela 8 apresenta um resumo das métricas obtidas na época 28.

Tabela 8 – Resultados em termos de métrica ROC-AUC (%) obtidos na abordagem baseadas em segmentação semântica com pré-treinamento.

	Treino	Validação
<i>Soft voting</i> sem máscara	88,56	63,35
<i>Hard voting</i> sem máscara	75,77	61,04
<i>Soft voting</i> com máscara	88,49	64,41
<i>Hard voting</i> com máscara	77,48	61,79

Assim como na abordagem sem pré-treinamento, o melhor resultado obtido no conjunto de validação foi com a utilização de *soft voting* juntamente com a máscara do tumor. A Figura 48 apresenta a curva ROC obtida para essa abordagem, no conjunto de teste, a qual resultou em uma métrica ROC-AUC de 61,99%.

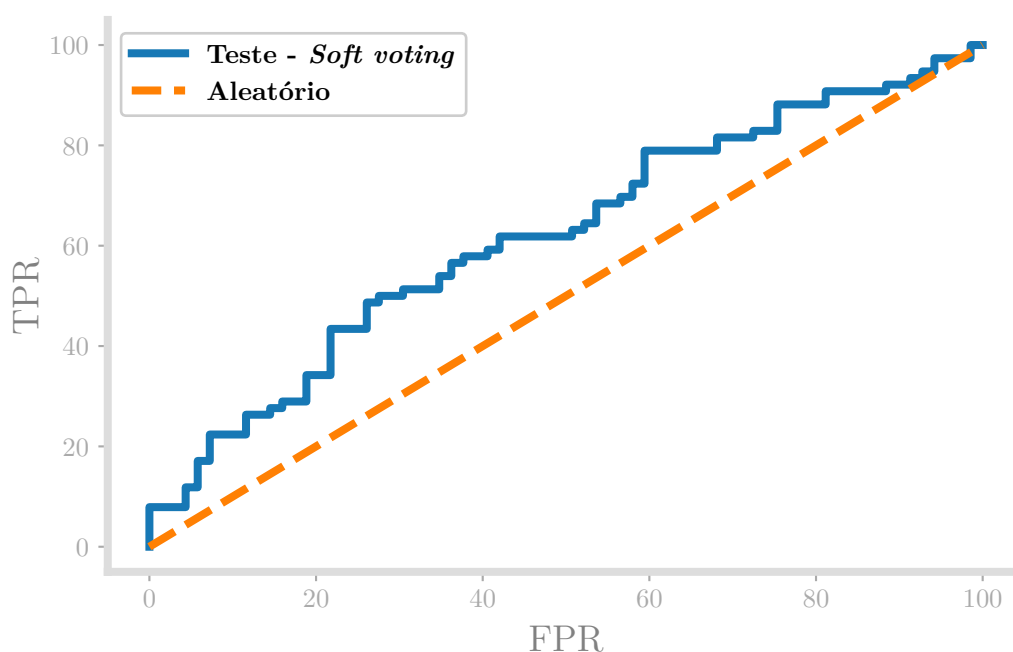


Figura 48 – Curvas ROC para a abordagem baseada em segmentação semântica.

### 5.4.3 Resultados no conjunto de dados TCIA

Os resultados obtidos no conjunto de dados BRATS 2021 além de não apresentarem melhoras com a utilização de um pré-treinamento na tarefa de classificação do *status* do gene IDH não chegaram a patamares próximos ao apresentado em (YOGANANDA *et al.*, 2021).

Devido a isso, essa abordagem foi reproduzida no conjunto de dados TCIA (o mesmo que o utilizado por (YOGANANDA *et al.*, 2021)), com o intuito de se identificar o impacto do conjunto de dados nos resultados obtidos. O modelo de segmentação semântica selecionado foi o mesmo utilizado anteriormente devido à falta de informações sobre a arquitetura utilizada pelos autores.

A Tabela 9 apresenta os resultados obtidos com técnica de votação *soft* devido a seu melhor desempenho:

Tabela 9 – Resultados em termos de ROC-AUC (%) obtidos na abordagem baseadas em segmentação semântica no conjunto de dados TCIA.

Técnica	Treino	Validação	Teste
<i>Soft voting</i> sem máscara - Sem pré-treino	52,92	55,86	56,63
<i>Soft voting</i> sem máscara - Com pré-treino	71,65	72,41	74,13

Nota-se que a utilização de um pré-treinamento conseguiu melhorar os resultados de ROC-AUC no conjunto de teste, porém, o resultado ainda ficou inferior ao apresentado por (YOGANANDA *et al.*, 2021). Alguns pontos ajudam a justificar essa diferença entre os resultados:

- a não disponibilização das sequências (de RMI) utilizadas para cada paciente criou uma dificuldade na reconstrução exata do conjunto de dados;
- devido à falta de informações sobre detalhes da arquitetura utilizada (número de filtros, tamanho do *kernel*, etc.) não foi possível reconstruir a arquitetura DenseU-Net utilizada pelos autores;
- os autores utilizaram uma etapa de pré-treinamento com validação cruzada de 3 *folds* a qual também foi utilizada pelo treinamento no conjunto de dados MGMT, não mantendo a consistência entre os conjuntos, isto é, pacientes utilizados para treinamento na tarefa de classificação do *status* do gene IDH foram utilizados para teste na tarefa referente ao gene MGMT. Isso causa um problema de vazamento de informação (*data leakage*) que pode resultar em um desempenho excessivamente otimista.

Mesmo sem atingir os mesmos patamares dos trabalhos anteriores a métrica ROC-AUC no conjunto de dados TCIA obteve valores mais elevados, demonstrando que a classificação automática tem uma maior capacidade de classificação nesse conjunto de dados, possivelmente devido à sua menor variabilidade em relação a um conjunto de dados maior.

## 5.5 COMPARAÇÃO DE ABORDAGENS E DISCUSSÃO

Todas as abordagens resultaram em métricas ROC-AUC em torno de 60%, diferentemente dos resultados obtidos em alguns dos trabalhos apresentados anteriormente, porém em conformidade com os resultados obtidos pelos participantes da competição BRATS 2021 e com (SAEED *et al.*, 2022).

A Tabela 10 apresenta os resultados obtidos em cada uma das abordagens. Os valores em negrito indicam que o valor foi o maior obtido. Pode-se notar a proximidade entre as métricas obtidas ao longo das diferentes abordagens, sendo a principal diferença observada nas métricas de treino.

Tabela 10 – Comparação em termos de ROC-AUC (%) dos resultados obtidos ao longo dos experimentos.

Abordagem	Treino	Validação	Teste
Extração de <i>features</i> - Regressão Logística	68,42	58,17	<b>65,02</b>
Extração de <i>features</i> - <i>Gradient boosting</i>	78,24	54,08	56,83
Extração de <i>features</i> - <i>Random forests</i>	76,53	55,43	60,00
<i>End-to-end</i>	71,60	62,26	58,46
Fatias - <i>Hard voting</i>	77,52	60,68	60,55
Segmentação semântica - <i>Soft voting</i> - Com máscara - Sem pré-treino	86,35	63,42	56,44
Segmentação semântica - <i>Soft voting</i> - Com máscara - Com pré-treino	88,49	<b>64,41</b>	61,99

O melhor resultado obtido foi o referente à abordagem via extração de *features*, utilizando um modelo de regressão logística para classificação dos pacientes. Um ponto de atenção é o desvio-padrão obtido nessa métrica (2,10), o qual ajuda a justificar esse desempenho superior.

A Figura 49 apresenta a comparação das curvas ROC para cada uma das abordagens presentes na Tabela 10. Essa comparação permite uma análise do desempenho dos classificadores em diferentes pontos de operação.



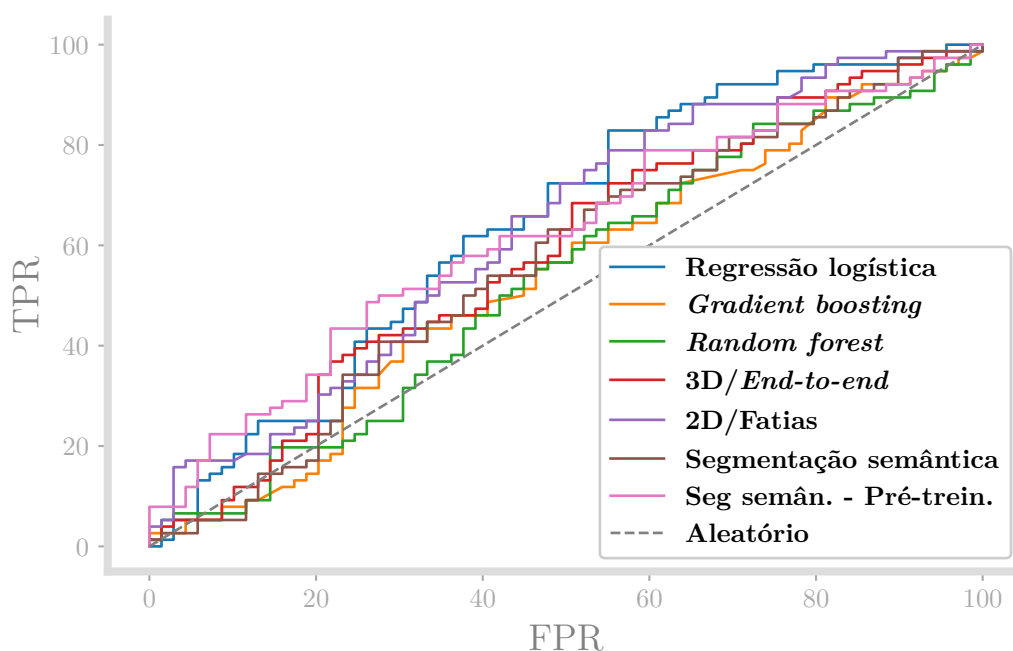


Figura 49 – Curvas ROC para os conjuntos de treino, validação e teste.

As métricas de teste no conjunto de dados BRATS 2021 evidenciam a dificuldade de se realizar a classificação automática do *status* do gene MGMT somente a partir de imagens de ressonância magnética.

Em termos gerais, todas as abordagens desenvolvidas têm diferentes necessidades, algumas delas com maior complexidade para sua implementação. Em termos de pré-processamento, uso da segmentação do tumor, *data augmentation* e tempo de treinamento, pode-se verificar o seguinte:

- o pré-processamento mais custoso é o aplicado na abordagem baseada em segmentação semântica de geração dos *patches* de treinamento, seguido pelo utilizado pela abordagem baseada em fatias. Na abordagem *end-to-end* nenhuma etapa de pré-processamento teve de ser utilizada devido a sua capacidade de lidar com a RMI completa;
- a necessidade do uso da segmentação de tumor para treinamento está presente em 3 das 4 abordagens testadas. Além da segmentação para treinamento a abordagem baseada em extração de *features* e a abordagem baseada em fatias necessitam da segmentação do tumor também para inferência. Isso faz com que os resultados sejam dependentes da qualidade da segmentação gerada, tornando as métricas apresentadas possivelmente mais otimistas que as reais;
- técnicas de *data augmentation* tiveram de ser utilizadas nas três abordagens baseadas em redes neurais convolucionais para evitar/reduzir o *overfitting* durante o treinamento. A aplicação de *data augmentation* “cancela” o efeito do co-registro,

sugerindo que, para abordagens baseadas em *deep learning*, a aplicação dessa técnica em conjuntos de dados pequenos acaba prejudicando o processo de treinamento dos modelos;

- o tempo de treinamento varia significativamente de abordagem para abordagem. Os experimentos baseados em extração de *features* tiveram os menores tempos de treinamento, facilitando a utilização de técnicas de validação cruzada mais robustas que a técnica *hold-out* e também permitindo uma melhor busca de hiperparâmetros. Já os modelos baseados em redes neurais convolucionais tiveram treinamentos com tempos variando de horas até dias.

## CAPÍTULO 6

---

### Conclusão

---

Nesse trabalho, a classificação do *status* do gene MGMT via utilização de aprendizado de máquina foi realizada, com base em quatro diferentes abordagens, visando a identificação de qual a melhor forma para resolução desse problema a partir da comparação de diferentes experimentos em uma base de dados comum, pública e de elevada qualidade.

A classificação da metilação do gene MGMT auxilia na construção de tratamentos mais eficazes, principalmente os baseados em Temozolomida (um quimioterápico comumente utilizado), o qual complementa os resultados cirúrgicos, melhorando tanto o tempo de sobrevivência dos pacientes quanto sua qualidade de vida (TRINDADE *et al.*, 2012).

Além da própria utilidade médica desse tipo de solução, o interesse nessa tarefa de classificação também ocorre devido à variabilidade dos resultados presentes na literatura e dos resultados obtidos durante a competição BRATS 2021. A falta de comparação de diferentes abordagens em uma base de dados comum, bem como a diferença entre as bases de dados utilizadas, dificultaram a identificação de qual a melhor técnica (caso exista uma) para resolução desse problema.

Nesta dissertação, os resultados apresentados mostram que métodos não-invasivos, baseados em aprendizado de máquina, para classificação do *status* do gene MGMT apresentam, em média, uma ROC-AUC (%) de 60% ( $\pm 2,78\%$ ). Esses resultados ficam em linha com os apresentados em (SAEED *et al.*, 2022) e com os resultados obtidos na competição BRATS 2021, mostrando a dificuldade de se obter bons diagnósticos automatizados da metilação do gene MGMT somente com a utilização desse tipo de dado.

Mesmo com resultados de métrica ROC-AUC inferiores aos da literatura e pessimistas em relação à utilização de métodos de aprendizado de máquina para classificação do *status* do gene MGMT, a comparação de diferentes abordagens em um conjunto de dados comum e com elevado número de pacientes (em relação aos conjuntos de dados públicos disponíveis) serviu

para realizar uma avaliação justa das diferentes técnicas mapeadas, eliminando as diferenças de resultados oriundas da diferença entre as bases de dados.

O principal ponto a ser abordado para evolução das técnicas de automação do diagnóstico do gene MGMT, é em relação à utilização de conjuntos de dados maiores e com significativa variabilidade para desenvolvimento das soluções. A medida que conjunto de dados maiores são utilizados, os resultados tendem a se tornarem mais representativos do real impacto que técnicas de I.A. tem na solução desse problema.

## 6.1 TRABALHOS FUTUROS

Como trabalhos futuros algumas possibilidades podem ser abordadas:

- análise baseada em métodos de explicabilidade dos resultados obtidos;
- teste de novas arquiteturas de redes neurais convolucionais;
- previsão do tempo de sobrevida de pacientes com base em imagens de ressonância magnética.

---

## REFERÊNCIAS

---

ALMEIDA, Marina Longo Machado de; AGUIAR, Paulo Henrique Pires de; GOIS, Katharyna de; SOUSA GEHRKE, Flavia de; FONSECA, Fernando. The importance of MGMT Promoter Methylation Status for Glioblastomas Prognosis: meta-analysis.

**JBNC-JORNAL BRASILEIRO DE NEUROCIRURGIA**, v. 29, n. 4, p. 595–611, 2019.

BAID, Ujjwal *et al.* The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification. **arXiv preprint arXiv:2107.02314**, 2021.

BANGALORE YOGANANDA, Chandan Ganesh *et al.* A novel fully automated MRI-based deep-learning method for classification of IDH mutation status in brain gliomas.

**Neuro-oncology**, Oxford University Press US, v. 22, n. 3, p. 402–411, 2020.

BASODI, Sunitha; JI, Chunyan; ZHANG, Haiping; PAN, Yi. Gradient amplification: An efficient way to train deep neural networks. **Big Data Mining and Analytics**, TUP, v. 3, n. 3, p. 196–207, 2020.

BLOCH, Felix. Nuclear induction. **Physical review**, APS, v. 70, n. 7-8, p. 460, 1946.

BUSLAEV, Alexander; IGLOVIKOV, Vladimir I.; KHVEDCHENYA, Eugene; PARINOV, Alex; DRUZHININ, Mikhail; KALININ, Alexandr A. Alumentations: Fast and Flexible Image Augmentations. **Information**, v. 11, n. 2, 2020. ISSN 2078-2489.

CHANG, P *et al.* Deep-learning convolutional neural networks accurately classify genetic mutations in gliomas. **American Journal of Neuroradiology**, Am Soc Neuroradiology, v. 39, n. 7, p. 1201–1207, 2018.

CHEN, Tianqi; HE, Tong; BENESTY, Michael; KHOTILOVICH, Vadim; TANG, Yuan; CHO, Hyunsu; CHEN, Kailong *et al.* Xgboost: extreme gradient boosting. **R package version 0.4-2**, v. 1, n. 4, p. 1–4, 2015.

DENG, Jia; DONG, Wei; SOCHER, Richard; LI, Li-Jia; LI, Kai; FEI-FEI, Li. ImageNet: A large-scale hierarchical image database. *In: 2009 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2009. P. 248–255.

DUBITZKY, Werner; GRANZOW, Martin; BERRAR, Daniel P. **Fundamentals of data mining in genomics and proteomics**. [S.l.]: Springer Science & Business Media, 2007.

FATIMA, Anam; SHAHID, Ahmad Raza; RAZA, Basit; MADNI, Tahir Mustafa; JANJUA, Uzair Iqbal. State-of-the-art traditional to the machine-and deep-learning-based skull stripping techniques, models, and algorithms. **Journal of Digital Imaging**, Springer, v. 33, n. 6, p. 1443–1464, 2020.

FAWCETT, Tom. An introduction to ROC analysis. **Pattern recognition letters**, Elsevier, v. 27, n. 8, p. 861–874, 2006.

GEPP, Ricardo de Amoreira. Análise prognóstica comparativa entre a perfusão cerebral por ressonância magnética e análise da metilação do gene promotor de MGMT nos gliomas malignos, 2013.

GÉRON, Aurélien. **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow**. [S.l.]: Alta Books, 2019.

GLIOBLASTOMA, NEWLY DIAGNOSED. Glioblastoma multiforme recém-diagnosticado: diagnóstico e tratamento cirúrgico inicial. **Acta Médica**, v. 39, n. 2, p. 164, 2018.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

GOYAL, Manu; GUO, Junyu; HINOJOSA, Lauren; HULSEY, Keith; PEDROSA, Ivan. Automated kidney segmentation by mask R-CNN in T2-weighted magnetic resonance imaging. *In: SPIE. MEDICAL Imaging 2022: Computer-Aided Diagnosis*. [S.l.: s.n.], 2022. v. 12033, p. 789–794.

HAGE, Maria Cristina Ferrarini Nunes Soares; IWASAKI, Masao. Imagem por ressonância magnética: princípios básicos. **Ciência Rural**, SciELO Brasil, v. 39, p. 1275–1283, 2009.

HAMGHALAM, Mohammad; LEI, Baiying; WANG, Tianfu. Brain tumor synthetic segmentation in 3D multimodal MRI scans. *In: SPRINGER. INTERNATIONAL MICCAI Brainlesion Workshop. [S.l.: s.n.], 2019. P. 153–162.*

HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. Deep residual learning for image recognition. *In: PROCEEDINGS of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2016. P. 770–778.*

HUANG, Gao; LIU, Zhuang; VAN DER MAATEN, Laurens; WEINBERGER, Kilian Q. Densely connected convolutional networks. *In: PROCEEDINGS of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2017. P. 4700–4708.*

IGLESIAS, Juan Eugenio; LIU, Cheng-Yi; THOMPSON, Paul M; TU, Zhuowen. Robust brain extraction across datasets and comparison with publicly available methods. **IEEE transactions on medical imaging**, IEEE, v. 30, n. 9, p. 1617–1634, 2011.

ISENSEE, Fabian; JAEGER, Paul F; KOHL, Simon AA; PETERSEN, Jens; MAIER-HEIN, Klaus H. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. **Nature methods**, Nature Publishing Group, v. 18, n. 2, p. 203–211, 2021.

ISENSEE, Fabian; JÄGER, Paul F; FULL, Peter M; VOLLMUTH, Philipp; MAIER-HEIN, Klaus H. nnU-net for brain tumor segmentation. *In: SPRINGER. INTERNATIONAL MICCAI Brainlesion Workshop. [S.l.: s.n.], 2020. P. 118–132.*

ISENSEE, Fabian *et al.* Automated brain extraction of multisequence MRI using artificial neural networks. **Human brain mapping**, Wiley Online Library, v. 40, n. 17, p. 4952–4964, 2019.

JOHNSON, Derek R; O'NEILL, Brian Patrick. Glioblastoma survival in the United States before and during the temozolomide era. **Journal of neuro-oncology**, Springer, v. 107, n. 2, p. 359–364, 2012.

KE, Guolin; MENG, Qi; FINLEY, Thomas; WANG, Taifeng; CHEN, Wei; MA, Weidong; YE, Qiwei; LIU, Tie-Yan. Lightgbm: A highly efficient gradient boosting decision tree. **Advances in neural information processing systems**, v. 30, 2017.

KINGMA, Diederik P; BA, Jimmy. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

KLEESIEK, Jens; URBAN, Gregor; HUBERT, Alexander; SCHWARZ, Daniel; MAIER-HEIN, Klaus; BENDSZUS, Martin; BILLER, Armin. Deep MRI brain extraction: A 3D convolutional neural network for skull stripping. **NeuroImage**, Elsevier, v. 129, p. 460–469, 2016.

KORFIATIS, Panagiotis; KLINE, Timothy L; LACHANCE, Daniel H; PARNEY, Ian F; BUCKNER, Jan C; ERICKSON, Bradley J. Residual deep convolutional neural network predicts MGMT methylation status. **Journal of digital imaging**, Springer, v. 30, n. 5, p. 622–628, 2017.

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. **Advances in neural information processing systems**, v. 25, 2012.

LE, Nguyen Quoc Khanh; DO, Duyen Thi; CHIU, Fang-Ying; YAPP, Edward Kien Yee; YEH, Hui-Yuan; CHEN, Cheng-Yu. XGBoost improves classification of MGMT promoter methylation status in IDH1 wildtype glioblastoma. **Journal of personalized medicine**, Multidisciplinary Digital Publishing Institute, v. 10, n. 3, p. 128, 2020.

MONAI CONSORTIUM. **MONAI: Medical Open Network for AI**. [S.l.: s.n.], fev. 2022. Disponível em: <https://github.com/Project-MONAI/MONAI>.

PASZKE, Adam *et al.* PyTorch: An Imperative Style, High-Performance Deep Learning Library. *In*: WALLACH, H.; LAROCHELLE, H.; BEYGELZIMER, A.; D'ALCHÉ-BUC, F.; FOX, E.; GARNETT, R. (Ed.). **Advances in Neural Information Processing Systems 32**. [S.l.]: Curran Associates, Inc., 2019. P. 8024–8035.

PEDREGOSA, F. *et al.* Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PROKHORENKOVA, Liudmila; GUSEV, Gleb; VOROBEV, Aleksandr; DOROGUSH, Anna Veronika; GULIN, Andrey. CatBoost: unbiased boosting with categorical features. **Advances in neural information processing systems**, v. 31, 2018.

RONNEBERGER, Olaf; FISCHER, Philipp; BROX, Thomas. U-net: Convolutional networks for biomedical image segmentation. *In*: SPRINGER. INTERNATIONAL Conference on Medical image computing and computer-assisted intervention. [S.l.: s.n.], 2015. P. 234–241.



SAEED, Numan; HARDAN, Shahad; ABUTALIP, Kudaibergen; YAQUB, Mohammad. Is it Possible to Predict MGMT Promoter Methylation from Brain Tumor MRI Scans using Deep Learning Models? **arXiv preprint arXiv:2201.06086**, 2022.

SINGH, Gagandeep *et al.* Radiomics and radiogenomics in gliomas: a contemporary update. **British Journal of Cancer**, Nature Publishing Group, v. 125, n. 5, p. 641–657, 2021.

SRIVASTAVA, Nitish; HINTON, Geoffrey; KRIZHEVSKY, Alex; SUTSKEVER, Ilya; SALAKHUTDINOV, Ruslan. Dropout: a simple way to prevent neural networks from overfitting. **The journal of machine learning research**, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.

SUN, Li; ZHANG, Songtao; CHEN, Hang; LUO, Lin. Brain tumor segmentation and survival prediction using multimodal MRI scans with deep learning. **Frontiers in neuroscience**, Frontiers, p. 810, 2019.

TAN, Mingxing; LE, Quoc. Efficientnet: Rethinking model scaling for convolutional neural networks. *In*: PMLR. INTERNATIONAL conference on machine learning. [S.l.: s.n.], 2019. P. 6105–6114.

TRINDADE, Vinicius; PICARELLI, Helder; FIGUEIREDO, Eberval Gadelha; TEIXEIRA, Manoel Jacobsen. Gliomas: marcadores tumorais e prognóstico. **Arquivos Brasileiros de Neurocirurgia: Brazilian Neurosurgery**, Thieme Revinter Publicações Ltda, v. 31, n. 02, p. 91–94, 2012.

VAN GRIETHUYSEN, Joost JM *et al.* Computational radiomics system to decode the radiographic phenotype. **Cancer research**, AACR, v. 77, n. 21, e104–e107, 2017.

WANG, Feifan; JIANG, Runzhou; ZHENG, Liqin; MENG, Chun; BISWAL, Bharat. 3d u-net based brain tumor segmentation and survival days prediction. *In*: SPRINGER. INTERNATIONAL MICCAI Brainlesion Workshop. [S.l.: s.n.], 2019. P. 131–141.

WATT, Jeremy; BORHANI, Reza; KATSAGGELOS, Aggelos K. **Machine learning refined: Foundations, algorithms, and applications**. [S.l.]: Cambridge University Press, 2020.

YOGANANDA, CGB *et al.* MRI-based deep-learning method for determining glioma MGMT promoter methylation status. **American Journal of Neuroradiology**, Am Soc Neuroradiology, v. 42, n. 5, p. 845–852, 2021.

---

ZHANG, Xinyuan; FENG, Yanqiu; CHEN, Wufan; LI, Xin; FARIA, Andreia V; FENG, Qianjin; MORI, Susumu. Linear registration of brain mri using knowledge-based multiple mediator libraries. **Frontiers in neuroscience**, Frontiers Media SA, v. 13, p. 909, 2019.