

Grupo Gold Retriever

RSNA-MICCAI Brain Tumor Radiogenomic Classification

Projeto Final – Mineração de Dados Complexos

Danilo Silva de Albuquerque
Leandro Nascimento Gonçalves de Araújo
Luiza Lober de Souza Piva
Raysa Masson Benatti
Victor Leal de Almeida

Instituto de Computação - Unicamp, 17 de dezembro de 2021

Resumo

Com o objetivo de classificar a presença ou ausência da metilação de uma enzima conhecida como MGMT a partir de imagens de ressonância magnética, este trabalho foi desenvolvido utilizando múltiplas técnicas de redes neurais profundas para a obtenção dos indicadores desta característica. Aqui, estudou-se o impacto da classificação direta, da extração de features e da seleção de regiões de interesse realizadas pela rede. Os resultados obtidos foram analisados com auxílio da curva ROC, além das métricas loss e accuracy para diagnóstico de overfitting. Posteriormente, também realizou-se a comparação destas previsões com aquelas apresentadas na competição original disponibilizada pela plataforma Kaggle.

Sumário

1	Descrição do problema	2
2	Base de dados	3
2.1	Estrutura e pré-processamento	3
2.2	Seleção dentre as sequências	4
3	Implementando o modelo	5
3.1	Criação de uma baseline	5
3.2	Grad-CAM e explicabilidade	6
3.3	Extraindo features e utilizando SVM	7
3.4	O melhor modelo	8
4	Resultados	8
4.1	Análise da baseline	8
4.2	Aplicação da SVM	8
4.3	Análise do melhor modelo	9
4.4	Conclusões e melhorias possíveis	10

1 Descrição do problema

No contexto de tumores cerebrais, sabe-se que os diagnósticos que exigem testes invasivos – como é o caso das biópsias de tecidos para a análise genética de um tipo de tumor – não só podem ser problemáticos para os pacientes, como também consomem um tempo precioso para o tratamento de tais doenças. Desta forma, o desenvolvimento de novas metodologias para identificação constitui um avanço na qualidade de atendimento dos pacientes.

Imagens por ressonância magnética (ou MRI – *magnetic resonance imaging*), por padrão, são utilizadas como parte da estratégia diagnóstica desse conjunto de doenças. A Figura 1 mostra cortes dos resultados de quatro métodos de imageamento, conhecidos na área como *janelas* (ou *sequências*) de um dado exame.

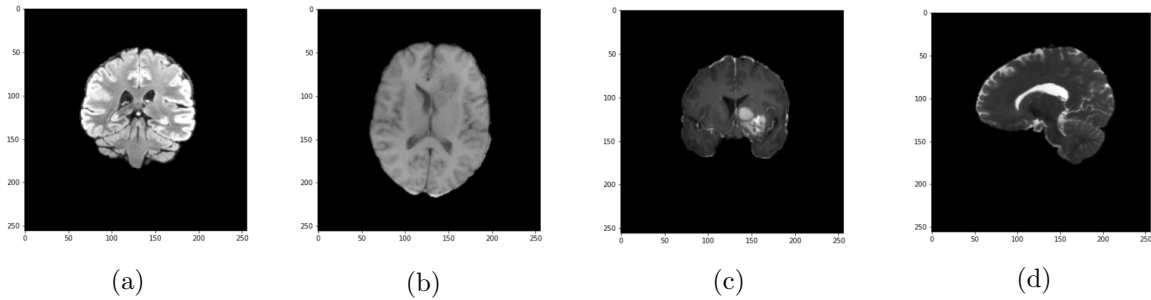


Figura 1: Resultados das quatro sequências de um exame de MRI para um dado paciente. (a) FLAIR. (b) T1w. (c) T1wCE. (d) T2w.

As principais características de cada uma dessas janelas são dadas pelo tipo de estrutura onde há um maior foco de intensidade luminosa. São elas,

1. **FLAIR**: a sigla significa “*Fluid Attenuated Inversion Recovery*”, ou seja, são imagens em que a contribuição dos fluidos cerebrais é reduzida no brilho total observado;
2. **T1w**: a gordura do tecido cerebral brilhará mais intensamente. O “w” significa *weighted*, explicitando que pesos são utilizados na distribuição de intensidades;
3. **T1wCE**: similar ao anterior, porém com uso de contraste (CE) para melhorias na resolução espacial da imagem;
4. **T2w**: qualquer fluido cerebral será ressaltado nessa janela.

Normalmente, um diagnóstico conclusivo requer ao menos duas dessas sequências, dado que a caracterização de um tumor se faz através da análise da distribuição de líquidos e padrões de gordura no cérebro.

Dito isto, um tipo específico de tumor, conhecido como glioblastoma multiforme (ou GBM, a evolução mais grave dos glioblastomas, classificada como grau quatro [1]), possui reações distintas a tratamentos quimioterápicos. Estes, por sua vez, têm forte influência da presença ou ausência de uma característica genética promotora da metilação de uma enzima conhecida como MGMT. O processo biológico completo é descrito em [2]. A metilação de MGMT tem a capacidade de reverter os danos causados no material genético tumoral pela quimioterapia.

Existe também a metilação aberrante no gene promotor da produção da MGMT, chamada *aberrant MGMT gene promoter methylation*, que reduz a habilidade do tumor de se regenerar do dano da quimioterapia. Desta forma, caso o paciente possua esta alteração genética, formas de extrair essa informação tornam-se indicadores favoráveis para tratamentos via quimioterapia para tumores que reajam bem a ela, o que é o caso para os glioblastomas. Desse ponto em diante, quando for citada a metilação do MGMT, é a esta que o texto se referirá.

Atualmente, a análise de DNA tumoral é feita somente com biópsia, mas trabalhos anteriores [3, 4] mostraram que é possível identificar padrões em imagens de ressonância magnética. Estes, por sua vez, podem se tornar bons classificadores para esta característica. Com isto em mente, a proposta do

projeto é que seja possível, a partir de estratégias baseadas em aprendizado de máquina, classificar a presença ou ausência da metilação de MGMT nesses exames, criando assim uma ferramenta não invasiva e rápida para auxiliar na tomada de decisão quanto ao tratamento da doença.

2 Base de dados

2.1 Estrutura e pré-processamento

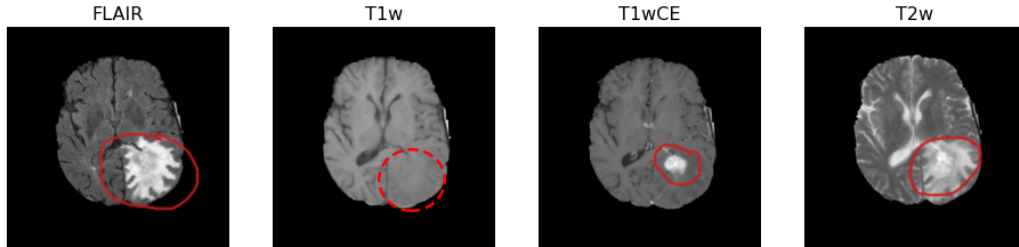


Figura 2: Sequências de um exame para um dado paciente. Nelas, $MGMT = 0$. Tanto nesta figura como na Figura 3, a região dos tumores [5] está circulada em vermelho.

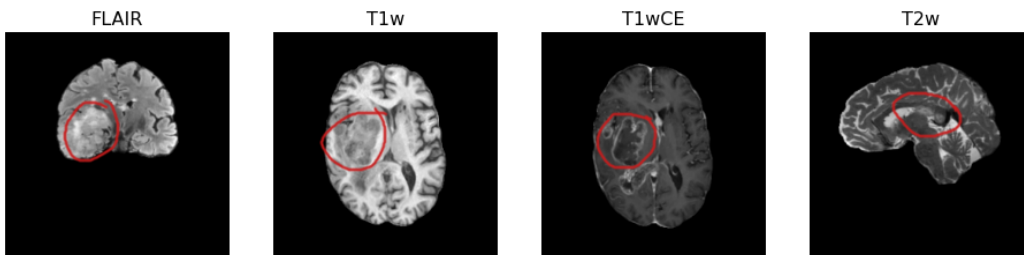


Figura 3: Sequências de um exame para outro paciente. Neste caso, $MGMT = 1$.

Aqui, a maneira como os dados estão estruturados no conjunto original da competição [6] será abordada.

- Pasta **train**: contém todas as amostras que serão utilizadas para o treinamento dos modelos. Trata-se de aproximadamente 400 mil imagens, referentes aos exames de 585 pacientes, cada um dos quais sendo representado por um número de identificação único.
 - Algumas amostras foram excluídas do treinamento, sendo elas as de identificação 00109, 00123 e 00709, conforme indicado nas discussões da competição [7]. O motivo são problemas de acesso à informação contida nas imagens – algumas sequências desses pacientes apresentavam apenas imagens escuras.
- Arquivo **train_labels.csv**: lista os valores de MGMT para os dados de treino, cujo domínio é binário em $\{0, 1\}$ – indicando ausência ou presença de MGMT, respectivamente.
- Pasta **test**: arquivos de teste, constituídos por imagens de exames de 87 pacientes. Estes arquivos só devem ser utilizados uma única vez para avaliar o melhor modelo.
 - Na competição do Kaggle, o tamanho total do conjunto de teste (público e privado) é 5x maior que a porção pública do conjunto. A submissão de novos trabalhos para a competição já havia sido encerrada antes do início desse projeto. Ademais, os rótulos mesmo da porção pública não foram disponibilizados em tempo hábil. Por essas razões, não foi possível usar o conjunto de teste original em nossas avaliações.

- Arquivo **sample_submission.csv**: uma amostra de rótulos para indicar o formato de saída esperado para a competição. O grupo adotou esta mesma formatação e as métricas de avaliação sugeridas, o que está descrito na Seção 3.1.
 - Esses arquivos são estruturados em duas colunas, com a primeira sendo **BraTS21ID**, representando o ID do paciente/exame, e a segunda o valor correspondente de **MGMT_value**. Ou seja, a saída deve prever uma probabilidade de existência do MGMT para cada paciente do conjunto de dados utilizado.

Note que as imagens do conjunto original de dados são em formato **dcm**, que permite o armazenamento de outras 55 informações no *header*, como tipo de sequência, rotação da figura, intensidade do campo magnético e posição do paciente. Apesar de permitir uma maior resolução das imagens e de conter as informações descritas, este formato exige grande poder de processamento para manipulação dos dados, o que adiciona uma camada a mais de dificuldade no treinamento inicial do modelo. Por este motivo, optamos por utilizar uma versão comprimida destes dados no formato **png**, disponibilizada em [8].

Com isto, estratificamos o conjunto de dados em porções para treino, validação e teste conforme descrito no Código 1: 15% dos dados de treino originais foram separados para teste e 20% do restante foram designados para validação. Esta escolha foi tomada para termos resultados comparáveis àqueles apresentados na competição do Kaggle.

```

1 df_trainval, df_test = sk_model_selection.train_test_split(
2     df,
3     test_size=0.15,
4     random_state=Config.SEED,
5     stratify=df["MGMT_value"],
6 )
7 df_train, df_val = sk_model_selection.train_test_split(
8     df_trainval,
9     test_size=0.2,
10    random_state=Config.SEED,
11    stratify=df_trainval["MGMT_value"],
12 )

```

Código 1: Separação estratificada do conjunto em treino, validação e teste

Vale dizer também que não observamos desbalanceamento significativo nas classes de MGMT, como mostrado abaixo na Figura 4.

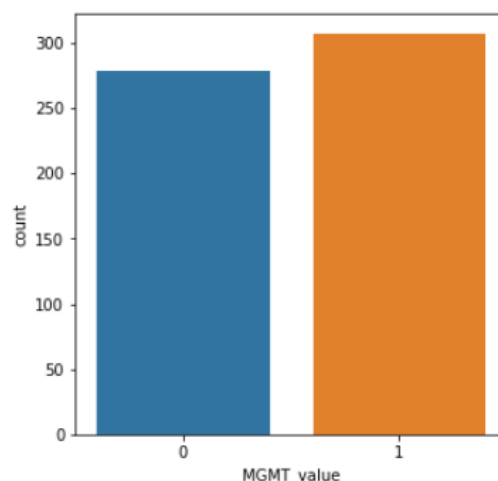


Figura 4: Contagem dos indicadores MGMT para os dados da base de treino.

2.2 Seleção dentre as sequências

Dado que combinar todas as sequências de uma só vez no modelo criaria uma camada a mais de complexidade em seu treinamento, optamos por definir quatro redes similares, cada uma sendo

treinada para uma dada sequência dos exames, como será abordado na Seção 3. Feito este treinamento, a probabilidade de presença do MGMT foi então tomada com uma função de máximo entre os valores de saída de cada uma das quatro redes.

Na baseline e nos experimentos com SVM, por sua vez, utilizamos somente sequências da categoria FLAIR. A escolha se justifica devido ao filtro aplicado para os fluidos da região cerebral – o que, em tese, facilita a identificação do tumor e de possíveis padrões de metilação de MGMT.

3 Implementando o modelo

3.1 Criação de uma baseline

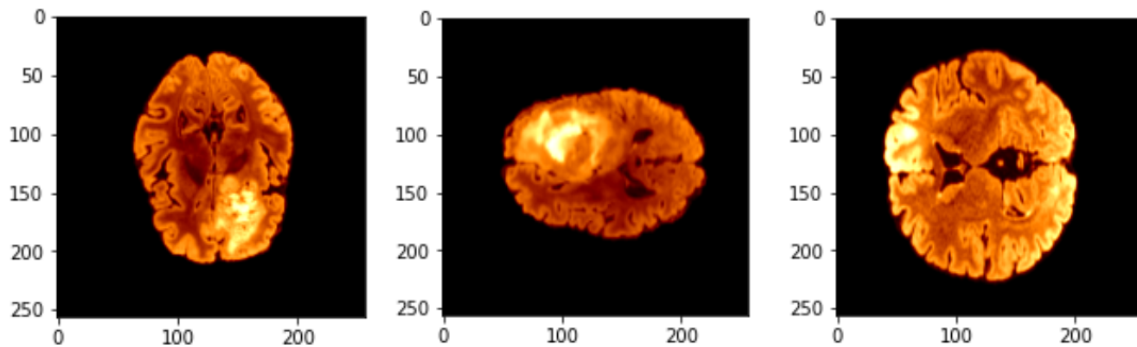


Figura 5: Resultado das funções de extração de dados listadas abaixo.

Especificamente como uma forma de definir um modelo inicial para este problema, a baseline contou com a implementação de algumas funções posteriormente utilizadas na classe `DataGenerator_2d_png`. No arquivo fonte, as linhas estão comentadas de forma a explicar, passo a passo, o que esta classe executa. São elas:

- `normalize_img`, colocando a imagem na escala de 0 a 255;
- `read_patient_png_2d`, responsável por selecionar o melhor corte dentro de uma sequência para um dado paciente através de um cálculo de rms, isto é, `rms = np.sqrt(np.mean(i_slice))`;
 - Note que, apesar de essa estratégia ser uma simplificação útil para o modelo, ela reduz significativamente o conjunto de dados que pode ser usado. Contando que somente imagens da janela FLAIR para 585 pacientes foram utilizadas, temos um total de 585 “melhores imagens” – que ainda passaram pela estratificação em treino, validação e teste, reduzindo a quantidade de imagens usadas para aprendizado efetivo. Para comparação, a ResNet foi treinada com a base de dados ImageNet, que possui 14 milhões de exemplos.
- A classe `DataGenerator_2d_png`, que produz os dataframes utilizados para o treino e validação do modelo. Aqui, define-se o tamanho do batch e a listagem das imagens a ser usadas através da função mencionada no último item.

Com isto feito, pudemos então implementar a rede neural de baseline para este problema. A solução via ResNet50v1 foi inicialmente escolhida por ser uma rede pré-treinada através da base de dados ImageNet e ser de grande utilização em problemas de reconhecimento e classificação de imagens [9]. Também foram adicionadas algumas modificações, contendo duas camadas densas finais, como mostrado no Código 2.

```
1 model_resnet = tf.keras.applications.ResNet50(weights='imagenet', include_top=False)
2
3 ipt = layers.Input(shape=(IMG_SIZE, IMG_SIZE, 1), name="input")
```

```

4 x = tf.keras.layers.Concatenate()([ipt, ipt, ipt])
5 x = tf.cast(x, tf.float32)
6 x = tf.keras.applications.resnet50.preprocess_input(x)
7 x = model_resnet(x)
8 x = layers.GlobalAveragePooling2D()(x)
9 x = layers.Dense(512, activation='relu')(x)
10 x = layers.Dropout(0.1)(x)
11 out = layers.Dense(1, activation='sigmoid')(x)
12
13 model_ResNet50_2d = Model(inputs=ipt, outputs=out)
14 model_ResNet50_2d.summary()

```

Código 2: Implementação do modelo de baseline com a ResNet50v1

Os parâmetros de otimização e de compilação podem ser conferidos no código original, com resultados para este modelo sendo apresentados na Seção 4.1.

Posteriormente à apresentação parcial deste trabalho, foram testadas outras redes e versões da ResNet como possíveis melhorias deste modelo, sendo elas ResNet50v2, ResNet101v2, ResNet152v2 e EfficientNet.B0. Porém, a partir da comparação das métricas usadas e de alguns resultados do Grad-CAM (detalhados na Seção 3.2), concluímos que a ResNet50v1 ainda se manteve como melhor alternativa para a modelagem.

A métrica de escolha para avaliação de todos os modelos é a AUC-ROC (área abaixo da curva ROC), conforme indicado na especificação da competição. Além disso, também observamos a evolução da *loss* de cada modelo nos conjuntos de treino e validação, de forma a ter um diagnóstico de *overfitting*.

3.2 Grad-CAM e explicabilidade

A proliferação do uso de soluções baseadas em inteligência artificial em diversos setores da sociedade vem sendo acompanhada de demandas legítimas, por diferentes partes interessadas, a respeito da transparência e *accountability* de tais soluções [10]. Nesse contexto, surge o conceito de inteligência artificial explicável (XAI, *explainable artificial intelligence*), cuja área sistematiza técnicas, métodos e estudos voltados ao desenvolvimento de sistemas de inteligência artificial cujas decisões sejam interpretáveis por seres humanos [11].

Essa demanda por explicabilidade se mostra particularmente importante em setores de impacto crítico, como é o caso em serviços de saúde. Diferentes sistemas normativos, que elencam direitos do paciente preveem alguma versão do direito a receber informações claras sobre diagnósticos e tratamentos [12, 13]; a explicabilidade e inteligibilidade de sistemas usados nesse domínio têm sido entendidas como dimensões importantes desse direito [14].

Nesse cenário, pesquisadores e profissionais vêm propondo diferentes técnicas para proporcionar maior interpretabilidade a modelos que, de outra forma, seriam usados como caixas-pretas – isto é, sem que se pudesse explicar com clareza por que o sistema tomou determinada decisão. Uma dessas técnicas é o Grad-CAM (*Gradient-weighted Class Activation Mapping*) [15], e ela foi a escolhida para a utilização neste trabalho. Trata-se de uma estratégia baseada em gradiente para gerar explicações visuais sobre decisões de classificação tomadas por redes neurais convolucionais.

A técnica usa as informações de gradiente direcionadas à última camada convolucional da rede para atribuir valores de importância a cada neurônio de uma decisão de interesse. Com isso, é possível inferir quão relevante cada região das imagens de entrada é para a decisão final de classificação. A abordagem permite que se produzam mapas de calor sobre as imagens, em que a temperatura da cor é proporcional à importância da respectiva região de *pixels* no aprendizado. Assim, a interpretabilidade vem dessa visualização.

A Figura 6, extraída do trabalho original, ilustra o que seria o funcionamento ideal da ferramenta. Nela, à esquerda, vemos a imagem original de um cachorro e um gato, fornecida como entrada para uma rede neural convolucional; à direita, o mapa de calor gerado mostra em vermelho e alaranjado as regiões de maior importância para a classificação da categoria *cachorro*. A cor esfria proporcionalmente ao afastamento dessa região. A visualização nos permite interpretar que, de fato, a decisão do modelo foi predominantemente baseada na porção da imagem que contém o cachorro em si, em especial seu rosto.

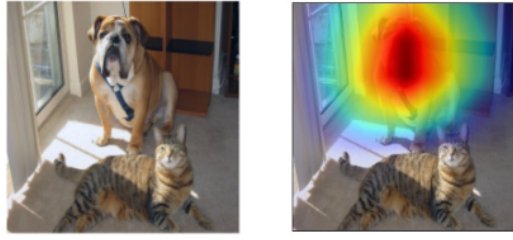


Figura 6: Ilustração do funcionamento ideal do Grad-CAM. Extraída e adaptada de Selvaraju et al. [15].

Há, contudo, ressalvas para seu uso no domínio médico, algumas das quais são discutidas no trabalho de Saporta et al. [16]. Em muitos cenários, a performance de explicabilidade desse grupo de ferramentas ainda é inferior à de experts de domínio, o que diminui sua confiabilidade; além disso, quando a técnica performa mal, é difícil estabelecer se isso se deve a problemas com o modelo escolhido, com a técnica em si ou com ambos. A própria interpretabilidade resultante da aplicação do método não é universal, dadas as diferenças institucionais e geográficas por trás da construção das habilidades de domínio necessárias para tal. Além disso, o desempenho da técnica pode variar significativamente entre diferentes modelos [17], o que inspira cautela em seu uso.

Ainda assim, consideramos importante estudar e aplicar a ferramenta neste trabalho, a fim de entendê-la melhor e explorar algum grau de explicabilidade nos nossos modelos. Sua primeira utilidade foi na escolha das arquiteturas de rede a explorar na etapa posterior ao baseline: comparando o desempenho do Grad-CAM na EfficientNet e na ResNet, vimos que a primeira teve mais dificuldade de aprender com as regiões corretas das imagens. A Figura 7 ilustra isso: à esquerda, no resultado sobre uma EfficientNet_B0, o calor está mal distribuído e difuso pela imagem toda; à direita, no resultado sobre uma ResNet50v1, o calor está mais concentrado e restrito à área do cérebro.

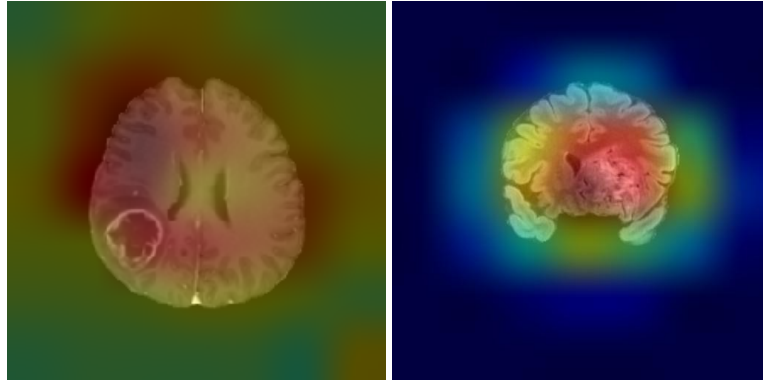


Figura 7: Resultado do Grad-CAM sobre duas redes distintas (EfficientNet à esquerda, ResNet à direita).

A técnica também foi usada para auxiliar na avaliação do desempenho do melhor modelo, conforme descrito na Seção 4.3.

3.3 Extraindo features e utilizando SVM

Como alternativa à classificação utilizando redes neurais, fizemos uma implementação de um método de classificação com SVM (*Support Vector Machines*) [18]. Para tanto, utilizamos a rede ResNet50v1 como extratora de informações das imagens, criando *feature vectors* (vetores de características). Estes vetores foram submetidos ao classificador SVM, construído com hiperparâmetros otimizados pela ROC AUC do conjunto de validação.

O processo de extração das características das imagens seguiu os mesmos padrões utilizados na construção do baseline, explicados na Seção 3.1. A especificidade, aqui, é a extração de features: a saída da ResNet50v1 gera um vetor de 2048 posições para cada imagem analisada. Como selecionamos

um slice da sequência FLAIR por paciente, o SVM recebeu como entrada um vetor de tamanho 2048 para cada caso de treino.

3.4 O melhor modelo

Além de algumas melhorias na organização do código quando comparado com a baseline – como a criação de uma classe para as configurações gerais do projeto (`class Config`) – a visualização de informação do conjunto de dados permaneceu a mesma em essência.

Uma diferença importante em relação à baseline é que, aqui, os dataframes foram montados com separações para cada tipo de janela do exame, o que pode ser observado no output de `df_val`. Esta mesma estrutura foi importante para a separação de cada uma dessas possibilidades de MRI para o treinamento de quatro diferentes modelos, um para cada janela. Essa iteração janela-a-janela é feita através da classe `PNGDataFrameIterator` e da função `get_iterating_dataframe()`.

Dito isto, a compilação dos modelos foi feita a partir de um loop `for` utilizando a lista definida com o nome de cada uma das janelas de MRI. Empregaram-se os pesos de treinamento da ImageNet, o congelamento das camadas da ResNet50 e a função `ModelCheckpoint()` para gravar os pesos e utilizá-los em futuras execuções da rede. Além disto, as mesmas camadas definidas para a baseline, mostradas no Código 2, foram aplicadas aqui, de onde por fim executou-se o treinamento do melhor modelo. Os resultados, posteriormente avaliados tanto para o conjunto de validação quanto para o de teste, são apresentados na Seção 4.3.

Vale dizer que, para este estágio final, o modelo de explicabilidade descrito na Seção 3.2 foi utilizado para um melhor entendimento de como a rede neural tomou suas decisões. Os resultados disto podem ser vistos nas Figuras 12 e 13.

4 Resultados

4.1 Análise da baseline

Dadas as implementações apresentadas na 3.1, obtivemos o resultado exposto na Figura 8.

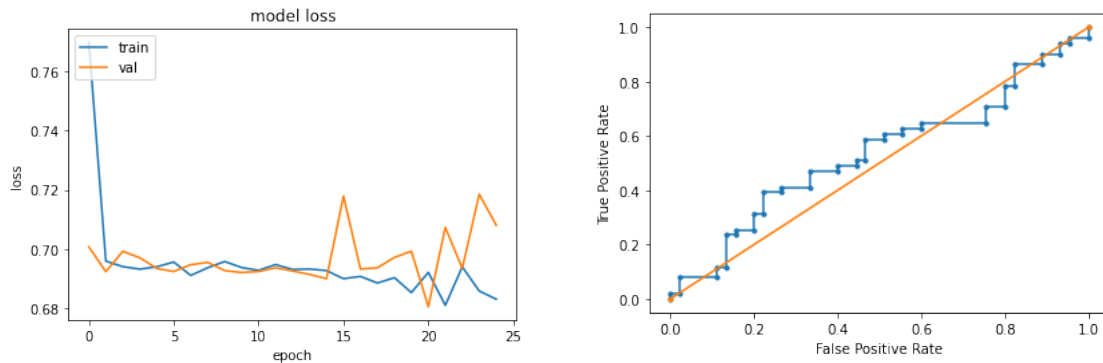


Figura 8: Resultados de (a) *loss* e (b) da área abaixo da curva ROC na validação para o modelo de baseline, aqui com $AUC = 0.535$.

Note que, apesar de estar ligeiramente acima do aleatório, os resultados do modelo de baseline não foi significativamente melhor do que este da Fig. 8. A abordagem utilizada ainda é bastante simplificada neste estágio, e em especial ao que tange ao tratamento dos dados que foram utilizados. Dado o baixo valor de AUC para todas as variações de janelas e de outras alterações no código, mesmo ao se utilizar uma divisão que permitisse um maior numero de imagens de treino para o modelo, como discutido na seção 2, optou-se por não aplicar este modelo no conjunto de teste.

4.2 Aplicação da SVM

A Figura 9 mostra os resultados obtidos com a estratégia de extração de features para os conjuntos de (a) treino, (b) validação e (c) teste. Apesar de as métricas estarem melhores que as da baseline,

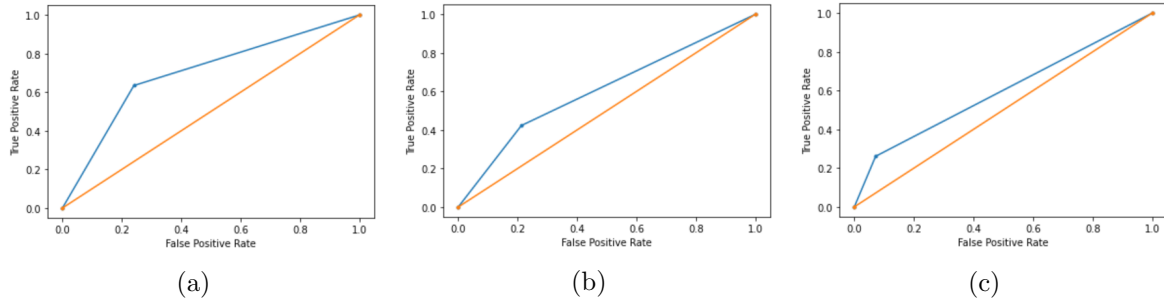


Figura 9: Resultados para o uso de extração de features no conjunto de (a) treino (AUC=0.697), (b) validação (AUC=0.605) e (c) teste (AUC=0.595). Note que a diferença entre validação e teste é pequena.

lembramos que aqui, também, foi usada somente a janela FLAIR, que tende a ter resultados melhores que as demais por padrão; além disso, o SVM usa uma quantidade de features (2048) maior que a de dados usados para treinamento (395). Isso tem como consequência gerar resultados abaixo do esperado ou fazer a rede tender ao *overfitting*, em especial para as demais janelas – o que pioraria uma possível combinação destas para um resultado completo do exame. Diante disso, optamos por não eleger essa estratégia como melhor modelo.

4.3 Análise do melhor modelo

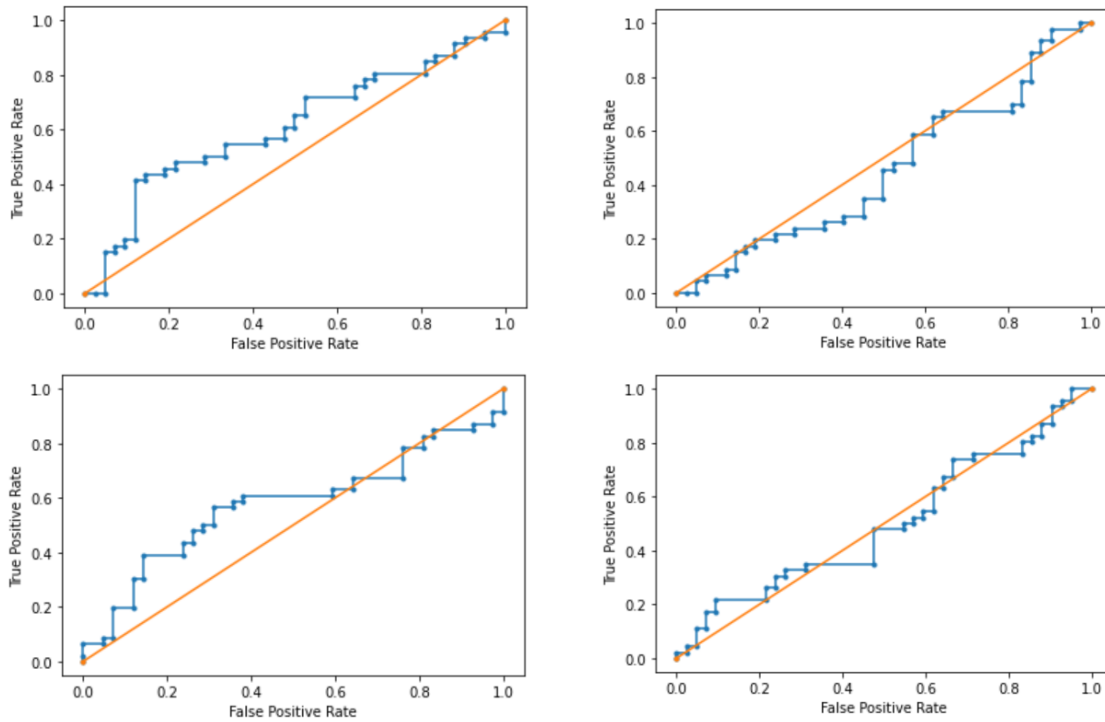


Figura 10: Resultados para o conjunto de teste da área abaixo da curva ROC nas janelas (a) FLAIR (AUC=0.612), (b) T1w (AUC=0.455), (c) T1wCE (AUC=0.575) e (d) T2w (AUC=0.502).

Assim como observado para a baseline, os valores de AUC no caso do melhor modelo não são muito diferentes da curva aleatória para todas as janelas apresentadas na Figura 10. Observe que os melhores resultados, assim como ocorreu em outras seções, é dado para a FLAIR. Isto pode ser justificado pela maior filtragem desta abordagem, como descrito na Sec. 1.

Por fim, ao combinar os resultados das janelas em uma única curva, obtivemos o resultado mostrado na Figura 11. Note que, devido às diferentes respostas para cada rede, a combinação dos quatro resultados terá AUC mais baixa que o melhor apresentado, Figura 10(a).

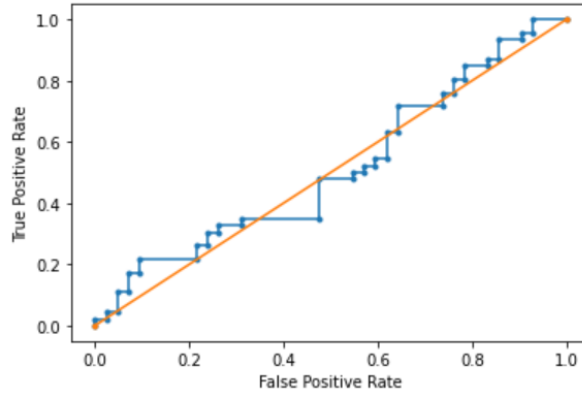


Figura 11: Combinação dos resultados das Figs. 10 em uma única curva ROC. O valor resultante é $AUC = 0.513$.

Nessa etapa, também aplicou-se a técnica Grad-CAM, cujos resultados destacamos as Figs. 12 e 13.

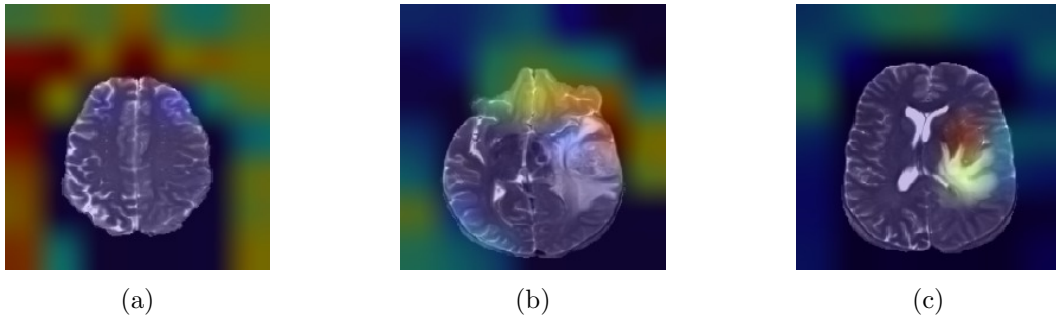


Figura 12: Decisões via Grad-CAM para cortes axiais usando a janela T2w. Note que há uma melhoria na indicação da posição do tumor ao se aproximar da possível localização dele, de (a) a (c).

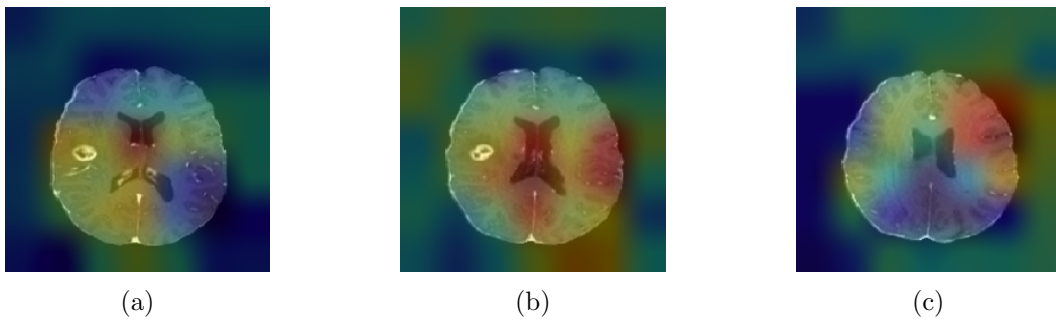


Figura 13: Resultados do Grad-CAM para o uso de outra janela (T1wCE) em outro paciente. Apesar de o tumor estar localizado na região central do lado esquerdo do cérebro [5], a rede não o localizou corretamente como na Figura 12.

Note que, a depender do paciente observado e da janela selecionada, a localização do glioblastoma pode ou não ser feita de forma precisa. Essa observada imprecisão, por consequência, irá afetar na determinação de padrões de metilação de MGMT, consequentemente piorando os resultados que são observados na curva ROC.

4.4 Conclusões e melhorias possíveis

Como exemplificado e discutido nas seções anteriores, este problema possui uma dificuldade intrínseca para o tratamento da dados. Correlacionar a existência de um tumor com um padrão que indique a presença ou ausência de metilação de MGMT para um dado paciente não só é uma tarefa árdua, como

requer um conjunto de treinamento extenso e padronizado, com as variáveis presentes – posição do paciente, tipo de corte, número total de imagens por janela, resolução, tamanho da imagem, dentre outras – configuradas de maneira similar para todos os dados [6]. Mesmo se isto fosse o caso para estes dados, ainda existe o fato de que o modelo procura por sinais de metilação de MGMT em imagens, mecanismo cujas possibilidades e funcionamento ainda não estão consolidados no domínio da neurologia – afinal, ainda não está totalmente definido quais são os padrões visuais que indicariam essa metilação. Uma discussão sobre este tema pode ser vista no site da própria competição, em [19], que conclui que provavelmente não é possível obter bons indicativos deste processo biológico somente por imageamento.

Outro ponto que requer melhorias é a quantidade de dados disponíveis. Apesar de a base original ter cerca de 400 mil imagens, elas vêm dos exames de apenas 585 pacientes; ao excluir slices com pouca ou nenhuma informação, o volume de dados efetivamente usado para treinamento diminui consideravelmente. Isso se torna ainda mais problemático considerando a qualidade discrepante da informação disponível em sequências diferentes, o que encolhe a quantidade de imagens adequadas para o aprendizado dos modelos. O problema poderia ser mitigado com uma estratégia robusta de aumento de dados e/ou o uso de redes 3D, que permitiriam o input dos dados em sua completude. Isso, contudo, demandaria poder de processamento maior que o que tivemos disponível.

Referências

- [1] Louis DN et al. The 2016 World Health Organization Classification of Tumors of the Central Nervous System: a summary. *Acta Neuropathol.*, 131(6):803–820, May 2016.
- [2] Nayak L and Reardon DA. High-grade Gliomas. *Continuum (Minneapolis)*, 23(6):1548–1563, December 2017.
- [3] Satrajit Chakrabarty, Aristeidis Sotiras, Mikhail Milchenko, Pamela LaMontagne, Michael Hileman, and Daniel Marcus. MRI-based Identification and Classification of Major Intracranial Tumor Types by Using a 3D Convolutional Neural Network: A Retrospective Multi-institutional Analysis. *Radiology: Artificial Intelligence*, 3(5):e200301, 2021.
- [4] Jaeyong Kang, Zahid Ullah, and Jeonghwan Gwak. MRI-Based Brain Tumor Classification Using Ensemble of Deep Features and Machine Learning Classifiers. *Sensors*, 21(6), 2021.
- [5] Anne G. Osborn, Karen L. Salzman, and Miral D. Jhaveri, editors. *Diagnostic Imaging: Brain*. Diagnostic Imaging. Elsevier, Philadelphia, third edition edition, 2016.
- [6] Ujjwal Baid et al. The RSNA-ASNR-MICCAI BraTS 2021 Benchmark on Brain Tumor Segmentation and Radiogenomic Classification, 2021.
- [7] Radiological Society of North America. RSNA-MICCAI Brain Tumor Radiogenomic Classification. <https://www.kaggle.com/c/rsna-miccai-brain-tumor-radiogenomic-classification>.
- [8] Jonathan Besomi. RSNA MICCAI’s competition data in PNG format (from 128 GB to 5.2 GB). <https://www.kaggle.com/jonathanbesomi/rsna-miccai-png>.
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition, 2015.
- [10] Ian Sample. Computer says no: why making AIs fair, accountable and transparent is crucial. *The Guardian*, 2017. <https://www.theguardian.com/science/2017/nov/05/computer-says-no-why-making-ais-fair-accountable-and-transparent-is-crucial>.
- [11] Waddah Saeed and Christian Omlin. Explainable AI (XAI): A Systematic Meta-Survey of Current Challenges and Future Opportunities, 2021.

- [12] Conselho Federal de Medicina. Código de Ética Médica – Resolução CFM 1931/09, 2009. <https://portal.cfm.org.br/images/stories/biblioteca/codigodeeticamedica.pdf>.
- [13] World Health Organization. A Declaration on the Promotion of Patients’ Rights in Europe, 1994. https://www.who.int/genomics/public/eu_declaration1994.pdf.
- [14] World Health Organization. Ethics and governance of artificial intelligence for health: WHO guidance, 2021. <https://www.who.int/publications/i/item/9789240029200>.
- [15] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision*, 128(2):336–359, Oct 2019.
- [16] Adriel Saporta, Xiaotong Gui, Ashwin Agrawal, Anuj Pareek, Steven QH Truong, Chanh DT Nguyen, Van-Doan Ngo, Jayne Seekins, Francis G. Blankenberg, Andrew Y. Ng, Matthew P. Lungren, and Pranav Rajpurkar. Benchmarking saliency methods for chest x-ray interpretation. *medRxiv*, 2021.
- [17] Siddhartha (Kaggle user). EfficientNet GradCam Comparison to Other Models, 2021. <https://www.kaggle.com/meaninglesslives/efficientnet-gradcam-comparison-to-other-models>.
- [18] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [19] John Mongan for the RSNA/MICCAI collaboration. On the low performance of models in this challenge, 2021. <https://www.kaggle.com/c/rsna-miccai-brain-tumor-radiogenomic-classification/discussion/284024>.