

LUCAS LEITE PINTO
LUCAS YUKIO TAMAI
PAULO OCTAVIO ARAÚJO DE PAULA

TREINAMENTO DE CLASSIFICADOR DE TIPO
HISTOLÓGICO DE NÓDULOS PULMONARES
COM DATASET SINTÉTICO GERADO POR
PGGAN - PROGRESSIVE GROWING GAN

São Paulo
2022

LUCAS LEITE PINTO
LUCAS YUKIO TAMAI
PAULO OCTAVIO ARAÚJO DE PAULA

**TREINAMENTO DE CLASSIFICADOR DE TIPO
HISTOLÓGICO DE NÓDULOS PULMONARES
COM DATASET SINTÉTICO GERADO POR
PGGAN - PROGRESSIVE GROWING GAN**

Trabalho apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro da Computação.

São Paulo
2022

LUCAS LEITE PINTO
LUCAS YUKIO TAMAI
PAULO OCTAVIO ARAÚJO DE PAULA

**TREINAMENTO DE CLASSIFICADOR DE TIPO
HISTOLÓGICO DE NÓDULOS PULMONARES
COM DATASET SINTÉTICO GERADO POR
PGGAN - PROGRESSIVE GROWING GAN**

Trabalho apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro da Computação.

Área de Concentração:
Engenharia da Computação

Orientador:
Prof. Dr. Edson Satoshi Gomi

Co-orientador:
Prof. Dr. Lucas Lins de Lima

São Paulo

2022

AGRADECIMENTOS

Agradecemos ao Professor Dr. Edson Gomi por nos aceitar como orientados e incentivar o grupo ao longo do projeto, bem como pelo interesse no tema e pela disponibilidade ao longo do ano.

Agradecimentos ao Professor Paulo Mazzoncini da Faculdade de Medicina de Ribeirão Preto (FMRP) da Universidade de São Paulo (USP) e a Lucas Lins Doutorando de Bioengenharia pela USP por sua colaboração e disponibilidade.

RESUMO

A detecção precoce do subtipo do câncer de pulmão é importante para o prognóstico e para decidir a linha de tratamento da doença, o que pode aumentar significativamente a taxa de sobrevivência dos pacientes. Atualmente, a classificação do subtipo é feita por meio de uma biópsia, um procedimento demorado, altamente invasivo e com potenciais implicações clínicas. Buscando contornar esses problemas, estudos recentes mostram que o uso de imagens de Tomografia Computadorizada aliadas com modelos de *Deep Learning* (DL) - especificamente redes neurais convolucionais - são muito efetivos para a classificação do tipo histológico dos nódulos. Todavia, o uso de modelos de DL apresenta outro desafio, a necessidade de grande quantidade de imagens anotadas por um especialista, o que no caso de imagens médicas é extremamente custoso. Para solucionar esse problema, esse trabalho propõe o uso de *Progressive Growing of Generative Adversarial Networks* para geração de imagens sintéticas adicionais que serão usadas para treinar um classificador e consequentemente melhorar seu desempenho. Concluímos que o uso das imagens geradas PGGAN melhora o desempenho do classificador quando comparado com o modelo treinado apenas com imagens reais, alcançando 0.940 de acurácia contra 0.906, respectivamente, uma melhora de 3,8%.

Palavras-Chave - Câncer do Pulmão, Tomografia Computadorizada, Aprendizagem de Máquina, Redes Generativas Adversariais

ABSTRACT

Early detection of the lung cancer subtype is essential for the prognosis and deciding the course of treatment for the disease, which can significantly increase the survival rate of patients. Currently, detection of the subtype is done through a biopsy, a time-consuming, highly invasive procedure with potential clinical implications. Seeking to circumvent these problems, recent studies show that using Computed Tomography images combined with Deep Learning (DL) algorithms, specifically convolutional neural networks, is very effective for classifying the histological type of nodules. However, the use of DL algorithms presents another complication: the need for a large number of images annotated by a specialist, which is highly costly in the case of medical images. To solve this problem, this work proposes using the Progressive Growing Generative Adversarial Networks to generate additional synthetic images that will be used to train the classifier and consequently improve its performance. We conclude that the use of PGGAN-generated images improves the performance of the classifier when compared to the model trained only with real images, reaching 0.940 of accuracy against 0.906, respectively, an improvement of 3.8%.

Keywords - Lung Cancer, Computed Tomography, Machine Learning, Generative Adversarial Networks

LISTA DE FIGURAS

- 1 Subtipos histopatológicos do câncer de pulmão em imagens de tomografia computadorizada: (a) Adenocarcinoma (ADC), (b) Carcinoma de células escamosas (SCC), (c) Carcinoma de grandes células (LCC). As caixas delimitadoras indicam a localização do tumor. Fonte: Autor 19
- 2 Algumas características de imagens de TC dos tumores: (A) Caso de ADC com margem de lobos não lisa, tumor lobulado e de formato irregular. (B) Caso de SCC com margem de lobos lisos e tumor de forma regular. Sinal de espessamento do feixe vascular em um caso de adenocarcinoma. (C) Vários pequenos vasos pulmonares foram tracionados e deslocados para a lesão. (D) Sinal de indentação pleural foi mostrado em um caso de SCC, mostrando uma extensão da lesão para a pleura e retração pleural adjacente. Fonte: [1] 20
- 3 Representação de uma imagem digital bidimensional. Fonte: [2] 21
- 4 Representação de um *voxel* e de um pixel: No (a) *voxel*, as dimensões do plano interno (Δx , Δy) e a espessura do corte no eixo z (Δz) são combinadas para formar o elemento de volume ou voxel; Como todas as imagens digitais, cada elemento da imagem de TC é chamado de (b) pixel. Fonte: [3] 22
- 5 Exemplo de (a) uma imagem de TC de pulmão e sua (b) representação em um histograma. Fonte: Autor 22
- 6 Aplicações de janelamento em TC de tórax: (a) janela de pulmão com largura de 1.400 UH e nível de -500 UH, (b) janela de mediastino com largura de 350 UH e nível de 40 UH, (c) janela de osso com largura de 1.000 UH e nível de 400 UH. Fonte: [4] 23

7	Esquema de Rede Neural Artificial. Fonte: [5]	24
8	Modelo do Neurônio Artificial.	25
9	Exemplo de convolução 2D. A saída foi restringida a somente posições em que o <i>kernel</i> é totalmente preenchido pela imagem, chamada de convolução "válida". Fonte: [6]	27
10	Exemplo de aplicação de <i>max pooling</i> de dimensão 2x2. Fonte: [7]	28
11	A arquitetura da VGG-16	29
12	Arquitetura da GAN: O objetivo do gerador (G) é gerar imagens que parecem o mais realistas possível, na tentativa de enganar o discriminador (D), um classificador que tenta classificar as imagens geradas como reais ou falsas. O objetivo do treinamento é alcançado quando D classifica as imagens geradas com 50% de confiança. Fonte: [8]	32
13	Arquitetura da PGGAN: No início, ambos G e D começam com uma dimensão de 4x4 e são treinados até a convergência. Na etapa seguinte, o treinamento é retomado com imagens com a dimensão duas vezes maior, 8x8, redimensionadas por um bloco de convolução. O processo se repete e as imagens aumentam gradualmente em escala por potências de 2 até que o tamanho necessário das imagens sintéticas seja alcançado (512x512). Fonte: [8]	34
14	Exemplo de segmentação de nódulos pulmonares do <i>dataset</i> HCFMRP: Imagens sequenciais de um paciente com (a) imagem de TC e (b) máscara de segmentação correspondente lado a lado. Fonte: [4]	39
15	Pré-processamento das imagens: (a) imagem de TC com tamanho 512 x 512, (b) ROI retangular centrada no nódulo com tamanho 94 x 81, (c) ROI normalizada e redimensionada com tamanho 64 x 64. Fonte: Autor	42

16	Exemplo de imagens sintetizadas pelo gerador em diferentes resoluções ao longo do treinamento da PGGAN, mostrando o aprendizado progressivo da rede. Fonte: Autor	44
17	Exemplo da sensibilidade do FID à distorções de imagem: (a) ruído gaussiano, (b) desfoque gaussiano, (c) inserção de retângulos pretos, (d) imagens espiraladas, (e) ruído de sal e pimenta e (f) <i>dataset</i> CelebA contaminado por imagens do <i>dataset</i> ImageNet. O nível de perturbação aumenta de zero para o nível mais alto. O FID capta bem o nível de perturbação crescendo monotonicamente. Fonte: [9]	48
18	Ilustração do GAN-train e GAN-test: O GAN-train aprende um classificador em imagens geradas pela GAN e mede o desempenho em imagens reais. Isso avalia a diversidade e o realismo das imagens geradas. O GAN-teste aprende um classificador com imagens reais e o avalia com imagens geradas pela GAN. Isso mede o quão realistas são as imagens geradas. Fonte: [10] .	50
19	Comparação entre imagens geradas pelas PGGANs e imagens reais para ADC: (a), (b), (c), (d), (e) e (f) são as imagens geradas pelas PGGANs na mesma ordem em que elas são apresentadas na Tabela 3, (g) são imagens reais. Fonte: Autor	51
20	Comparação entre imagens geradas pelas PGGANs e imagens reais para SCC: (a), (b), (c), (d), (e) e (f) são as imagens geradas pelas PGGANs na mesma ordem em que elas são apresentadas na Tabela 3, (g) são imagens reais. Fonte: Autor	52
21	Exemplo de curvas ROC: a curva 1 representa um teste com desempenho perfeito de área igual a 1; a curva 2 representa um teste com desempenho intermediário de área igual a 0,75; e a curva 3 é um teste aleatório com área igual a 0,5.	54

22	Avaliação da acurácia de validação para diferentes números de imagens geradas pela PGGAN que foram utilizadas no treinamento do classificador. A linha horizontal tracejada indica a acurácia do modelo treinado apenas com imagens reais. Em negrito estão os maiores valores por métrica. Fonte: Autor	56
23	Comparação entre curvas ROC para diferentes números de imagens geradas pela PGGAN que foram usadas no treinamento do classificador. A curva mais espessa representa o classificador treinado apenas com imagens reais. Fonte: Autor	57

LISTA DE TABELAS

1	Valores de HU de substâncias comuns.	22
2	Informações sobre as imagens contidas nos datasets utilizados	37
3	Avaliação das PGGANs treinadas com diferentes hiperparametros. Os valores em negrito representam o menor valor de FID ou os maiores valores dos escores GAN-test e GAN-train na coluna. Para fins comparativos, o valor da acurácia de validação para o modelo treinado e testado com imagens reais é 0,906. Fonte: Autor	51
4	Comparação das métricas de classificação para diferentes números de imagens sintéticas geradas pela PGGAN que foram juntamente com as imagens reais para o treinamento da VGG16. Em negrito estão os maiores valores por métrica por coluna.	55

LISTA DE ABREVIATURAS

ADC Adenocarcinoma

AUC Area under the curve

DICOM Digital Imaging and Communications in Medicine

FID Frechet Inception Distance

GAN Generative Adversarial Network

GPU Graphics processing unit

HU Hounsfield Units

LCC Large cell carcinoma

NSCLC Non-small cell lung cancer

PGGAN Progressive Growing of GANs

ROC Receiver Operating Characteristic Curve

ROI Region of interest

SCC Squamous cell carcinoma

TC Tomografía Computadorizada

XML eXtensible Markup Language

SUMÁRIO

1	Introdução	15
1.1	Objetivo	16
2	Aspectos Conceituais	18
2.1	Tipos e subtipos histológicos de câncer de pulmão	18
2.1.1	Correlações entre características de imagens de TC e histologia do câncer de pulmão	19
2.2	Visão Computacional	20
2.2.1	Imagem Digital	20
2.2.2	Classificação de Imagens	21
2.3	Tomografia Computadorizada	21
2.4	Redes Neurais Artificiais	23
2.4.1	Redes Neurais Convolucionais	26
2.4.2	VGG-16	28
2.5	Data Augmentation	29
2.5.1	Generative Adversarial Networks	30
2.5.2	Progressive Growing of GANs	32
3	Tecnologias Utilizadas	35
3.1	Python	35
3.1.1	Jupyter Notebook	35

3.1.2	NumPy	35
3.1.3	Matplotlib	36
3.2	PyTorch	36
3.3	Hardware	36
4	Materiais e Métodos	37
4.1	Datasets Utilizados	37
4.1.1	HCFMRP	37
4.1.2	Lung-PET-CT-Dx	39
4.1.3	NSCLC Radiomics	40
4.2	Pré-processamento das Imagens	40
5	Projeto e Implementação	43
5.1	Progressive Growing GAN	43
5.1.1	Detalhes de implementação	43
5.1.2	Treinamento da rede	43
5.2	VGG-16	44
5.2.1	Detalhes de implementação	44
5.2.2	Treinamento da rede	44
6	Avaliação	46
6.1	Avaliação da etapa de geração de imagens sintéticas	46
6.1.1	Métricas utilizadas	47
6.1.1.1	Fréchet Inception Distance (FID)	47

6.1.1.2	GAN-train e GAN-test	48
6.1.2	Resultados	50
6.2	Avaliação da etapa classificação de imagens	52
6.2.1	Métricas utilizadas	52
6.2.2	Resultados	54
7	Considerações Finais	58
7.1	Trabalhos Futuros	58
	Referências	59

1 INTRODUÇÃO

O câncer de pulmão é responsável por uma em cada cinco mortes por câncer e gerou custos diretos de mais de US\$ 823 milhões somente em 2016 no Brasil que ainda tem tido dificuldade em conseguir diagnosticar precocemente esse tipo de câncer [11].

O câncer de pulmão é classificado histopatologicamente e a categorização do tipo de malignidade com base na histopatologia é necessária para o prognóstico e para decidir a linha de tratamento [12]. Além disso, recentemente, o tratamento personalizado baseado no subtipo histológico do tumor e também em mutações genéticas tem levado ao aumento da sobrevida dos pacientes e traz grandes promessas para o futuro [13].

No entanto, é altamente desafiador identificar com precisão os tipos/subtipos histológicos de câncer de pulmão em termos de suas características morfológicas [12]. Normalmente, os pacientes são submetidos a uma biópsia pulmonar, em que uma amostra de tecido pulmonar é obtida e sua estrutura microscópica é analisada. A biópsia pulmonar em geral é altamente desafiadora, principalmente, para cânceres de pulmão localizados próximos a vias aéreas, vasos sanguíneos ou para aqueles localizados profundamente, o procedimento também é altamente invasivo com potenciais implicações clínicas, como infecção, hemorragia e danos aos tecidos circundantes, além disso, a biópsia nem sempre é eficaz para a determinação da histologia do câncer de pulmão [14].

Dessa forma, o uso de imagens de Tomografia Computadorizada para classificar tipos/subtipos histológicos de câncer de pulmão é uma forma de diagnóstico promissora por ser econômica, eficaz, não invasiva, comumente disponível e rápida [1][15][16], que pode ajudar os médicos a tomar decisões de tratamento direcionadas em tempo hábil e

também evitar que os pacientes sejam expostos aos riscos da biópsia pulmonar [12].

A inteligência artificial tem mostrado notável sucesso na análise de imagens médicas, e sistemas de suporte à decisão clínica baseados em modelos de IA são cada vez mais usados para auxiliar os médicos no processo de tomada de decisão [12]. Dentre esses modelos os de deep learning, em particular redes convolucionais, são o tipo de modelo mais bem-sucedido para análise de imagens até o momento [17].

Entretanto, o uso de modelos supervisionados de deep learning para classificação de imagens requer grande quantidade de dados anotados. Se tratando de imagens médicas isso é um grande problema, visto que, somente especialistas ou profissionais qualificados podem anotar os dados, e os custos associados à anotação de exemplos suficientes são drasticamente altos [18].

Para contornar esse problema um dos métodos mais adotados para aumentar o número de dados de treinamento é o *data augmentation*, cujas técnicas clássicas consistem na aplicação de um conjunto de transformações afins, como rotações, cortes e espelhamentos [19]. No entanto, essas imagens transformadas têm intrinsecamente uma distribuição semelhante às originais, levando a uma melhora de desempenho limitada. Assim, gerar amostras realistas (ou seja, semelhantes à distribuição da imagem real), mas completamente novas, é essencial para preencher a distribuição da imagem real descoberta pelo conjunto de dados original. Nesse contexto, o *data augmentation* baseado em Redes Generativas Adversárias (do Inglês *Generative Adversarial Networks* - GAN) é promissor, pois tem mostrado excelente desempenho em visão computacional, revelando boa capacidade de generalização [20].

1.1 Objetivo

O objetivo deste trabalho é treinar um classificador de tipo histológico de nódulos pulmonares em imagens de Tomografia Computadorizada (TC) usando modelos de *deep learning*. Devido à pequena quantidade de dados disponíveis para treinar o modelo,

espera-se também realizar *data augmentation* por meio da implementação de uma PG-GAN capaz de gerar imagens realistas, mas completamente novas, que são capazes de melhorar o desempenho e a robustez do modelo.

2 ASPECTOS CONCEITUAIS

2.1 Tipos e subtipos histológicos de câncer de pulmão

O câncer de pulmão configura-se como o crescimento descontrolado de células no parênquima pulmonar. Os cânceres de pulmão são classificados histopatologicamente como carcinoma pulmonar de células pequenas (do Inglês *small cell lung cancer* - SCLC) ou carcinoma pulmonar de células não pequenas (do Inglês *non-small cell lung cancer* - NSCLC), os SCLCs constituindo 15% e os NSCLCs constituindo 85% de toda a população de câncer de pulmão, respectivamente [21].

O NSCLC é ainda subdividido em três tipos: Adenocarcinoma (ADC, Figura 1(a)), Carcinoma de células escamosas (do Inglês *squamous cell carcinoma* - SCC, Figura 1(b)) e Carcinoma de células grandes (do Inglês *large cell carcinoma* - LCC, Figura 1(c)). Entre todos os cânceres de pulmão, o adenocarcinoma é o carcinoma de pulmão mais comum; contribuindo com 40% de todos os incidentes de câncer de pulmão, já o carcinoma de células escamosas contribui com 25 a 30% da incidência de câncer de pulmão, enquanto o carcinoma de células grandes compreende de 5 a 10% do total [21].

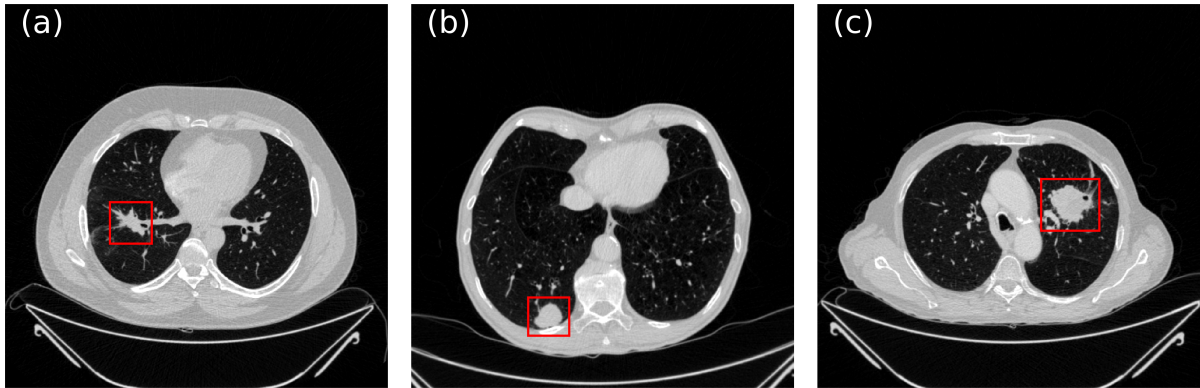


Figura 1: Subtipos histopatológicos do câncer de pulmão em imagens de tomografia computadorizada: (a) Adenocarcinoma (ADC), (b) Carcinoma de células escamosas (SCC), (c) Carcinoma de grandes células (LCC). As caixas delimitadoras indicam a localização do tumor. Fonte: Autor

2.1.1 Correlações entre características de imagens de TC e histologia do câncer de pulmão

Para que seja possível classificar corretamente os tipos e subtipos de câncer por meio de imagens de TC é importante que essas imagens contenham características que permitam distinguir um tipo de outro. Estudos mostram que algumas dessas características presentes em TCs de pulmão como, forma da lesão, margem dos lobos, se a lesão tinha margem lobulada profunda, espessamento do feixe vascular, sinais de dilatação das artérias brônquicas, indentação pleural, entre outras mostraram ter relação com a classificação histológica. Em comparação com SCC, os adenocarcinomas, por exemplo, são mais propensos a ter margem lobulada profunda, espessamento do feixe vascular e indentação pleural e menos propensos a ter margem de lobos lisos e artérias brônquicas dilatadas [15]. A Figura 2 mostra algumas das características presentes nas imagens de TC que permitem identificar o tipo histológico.

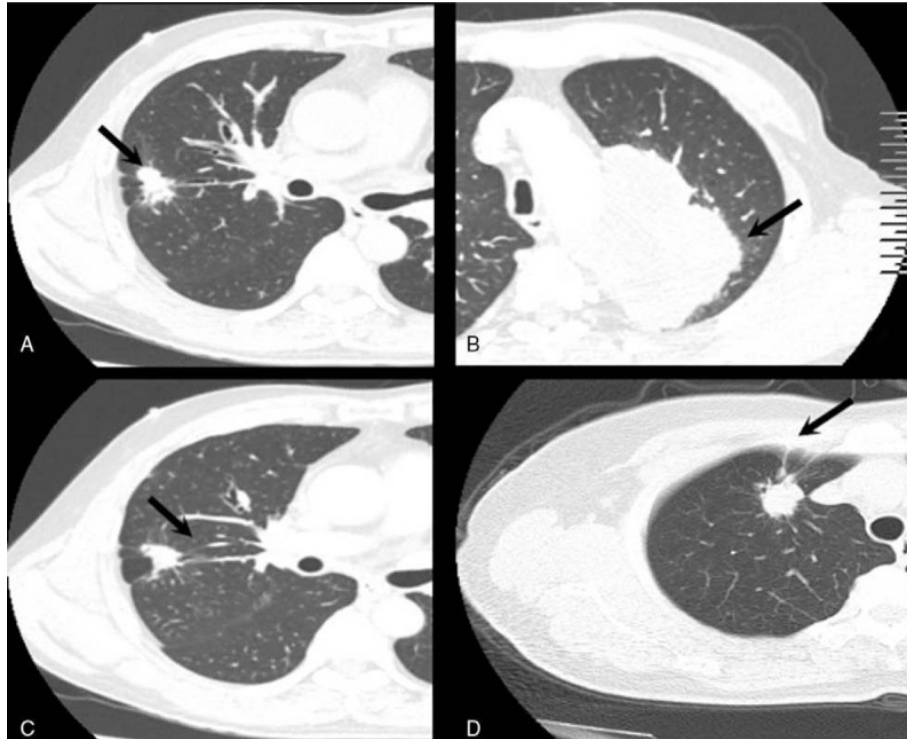


Figura 2: Algumas características de imagens de TC dos tumores: (A) Caso de ADC com margem de lobos não lisa, tumor lobulado e de formato irregular. (B) Caso de SCC com margem de lobos lisos e tumor de forma regular. Sinal de espessamento do feixe vascular em um caso de adenocarcinoma. (C) Vários pequenos vasos pulmonares foram tracionados e deslocados para a lesão. (D) Sinal de indentação pleural foi mostrado em um caso de SCC, mostrando uma extensão da lesão para a pleura e retração pleural adjacente. Fonte: [1]

2.2 Visão Computacional

2.2.1 Imagem Digital

Uma imagem digital é uma matriz 2D de pixels (Figura 3). Cada pixel é caracterizado por suas coordenadas (x, y) e seu valor de intensidade. As imagens digitais são caracterizadas pelo tamanho da matriz, pela resolução e pela profundidade de cor. O tamanho da matriz é determinado a partir do número de colunas (M) e do número de linhas (N) da matriz da imagem ($M \times N$). Para uma dada área de uma matriz, quanto maior o número de pixels melhor é a sua resolução espacial, o que permite um melhor diferenciação espacial entre as estruturas. A profundidade de cor refere-se ao número de bits por pixel que representam os níveis de cor de cada pixel em uma imagem. Cada pixel

pode assumir 2^k valores diferentes, onde k é a profundidade de cor da imagem. [2]

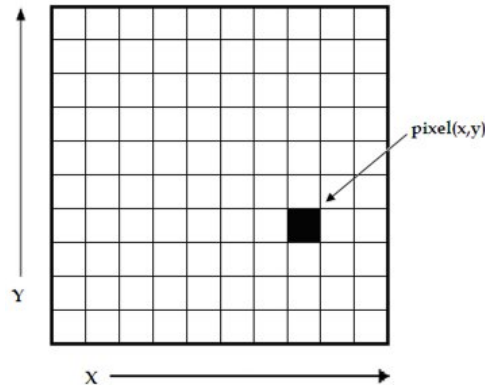


Figura 3: Representação de uma imagem digital bidimensional. Fonte: [2]

2.2.2 Classificação de Imagens

2.3 Tomografia Computadorizada

A Tomografia Computadorizada baseia-se no fato de que tecidos com diferentes composições absorvem a radiação X de forma diferente. Dessa forma, ao serem atravessados por raios X, tecidos mais densos (como o fígado) ou com elementos mais pesados (como o cálcio presente nos ossos), absorvem mais radiação que tecidos menos densos (como o pulmão).

Imagens de TC são comumente adquiridas em forma de volume de fatias paralelas e uniformemente espaçadas [4]. Imagens são renderizadas para que a visão seja como se estivesse olhando para ela a partir dos pés do paciente. Em uma imagem digital de TC cada pixel (Figura 4(b)) representa a atenuação média dos tecidos de um *voxel* (elemento de volume) no paciente. O *voxel* tem duas dimensões iguais ao pixel no plano da imagem, e a terceira dimensão representa a espessura do corte da tomografia (Figura 4(a)). Os pixels são convertidos para serem exibidos de acordo com o valor da atenuação média e a intensidade de cada pixel é expressa em uma escala cujas unidades são denominadas unidades Hounsfield (do Inglês *Hounsfield Units* - HU) (Figura 5) [3]. Os valores correspondentes a algumas substâncias na escala Hounsfield estão descritos na Tabela 1.

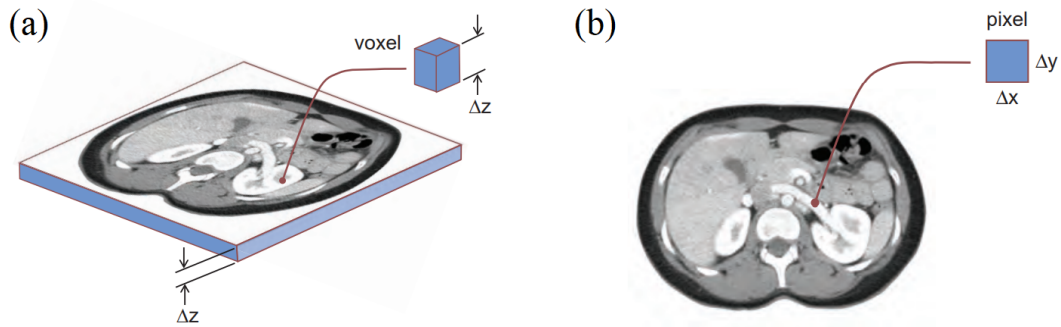


Figura 4: Representação de um *voxel* e de um pixel: No (a) *voxel*, as dimensões do plano interno (Δx , Δy) e a espessura do corte no eixo z (Δz) são combinadas para formar o elemento de volume ou *voxel*; Como todas as imagens digitais, cada elemento da imagem de TC é chamado de (b) pixel. Fonte: [3]

Substância	HU
Ar	-1000
Pulmão	-400 a -600
Gordura	-100 a -60
Água	0
Tecidos moles	+40 a +80
Osso	+1000

Tabela 1: Valores de HU de substâncias comuns.
Fonte: [22]

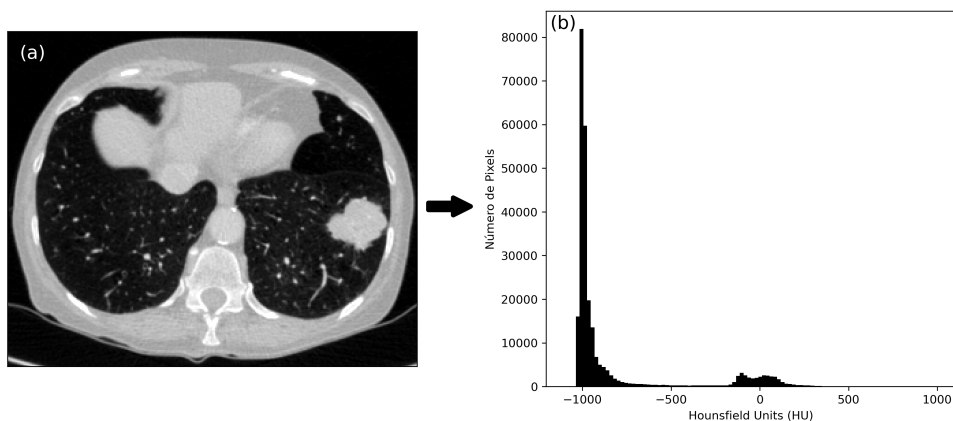


Figura 5: Exemplo de (a) uma imagem de TC de pulmão e sua (b) representação em um histograma. Fonte: Autor

Voxels de imagens de TC normalmente são representados por 12 bits em escala de cinza, totalizando 4.096 possíveis tons de cinza. Contudo, a sua escala possui grande

amplitude e a distribuição dos níveis de cinza entre os limites mínimo e máximo resulta em uma baixa diferenciação de contraste entre os tecidos [4]. Logo, é necessário realizar o janelamento da imagem de TC. A largura da janela (W) determina o contraste da imagem, e o nível (L) determina o valor no centro da janela. A seleção dos valores de L e W determina os pontos de inflexão $P1$ e $P2$, assim todos os números de CT inferiores a $P1$ são saturados para preto, e todos os números de CT acima de $P2$ são saturados para branco [3]. A Figura 6 mostra os efeitos do janelamento na visualização de diferentes tecidos em uma imagem de TC, para fins de padronização as imagens exibidas nesse trabalho estão na configuração do pulmão (largura da janela, 1,400 HU; nível, -500 HU).

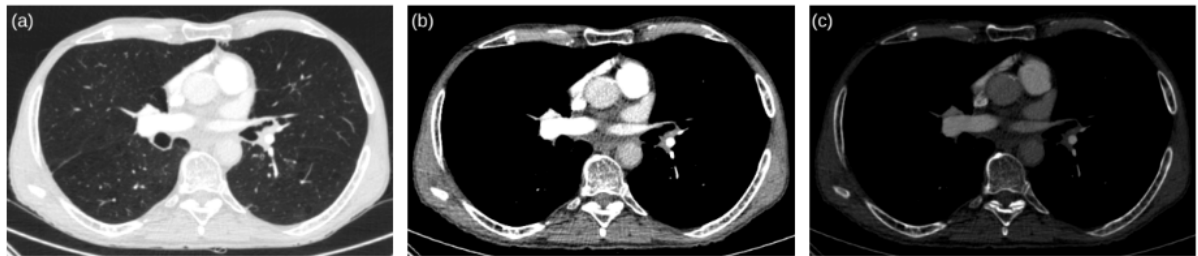


Figura 6: Aplicações de janelamento em TC de tórax: (a) janela de pulmão com largura de 1.400 UH e nível de -500 UH, (b) janela de mediastino com largura de 350 UH e nível de 40 UH, (c) janela de osso com largura de 1.000 UH e nível de 400 UH. Fonte: [4]

2.4 Redes Neurais Artificiais

Uma Rede Neural é um modelo computacional fortemente paralelizado constituído de simples unidades de processamento (neurônios, ou nós), que têm uma propensão natural para guardar conhecimento adquirido através de experiência e disponibilizar este conhecimento para uso. O modelo se assemelha ao cérebro de duas formas [23]:

1. O conhecimento é adquirido pela rede de seu ambiente por meio do aprendizado;
2. A força entre neurônios, chamada de peso, é usada para guardar o conhecimento adquirido;

Dessa forma, numa rede neural artificial, os neurônios se interconectam entre si através

de canais de comunicação, tendo cada um destes canais um peso associado a ele. Estes pesos são coeficientes que amplificam ou minimizam o sinal de entrada em um determinado neurônio na rede.

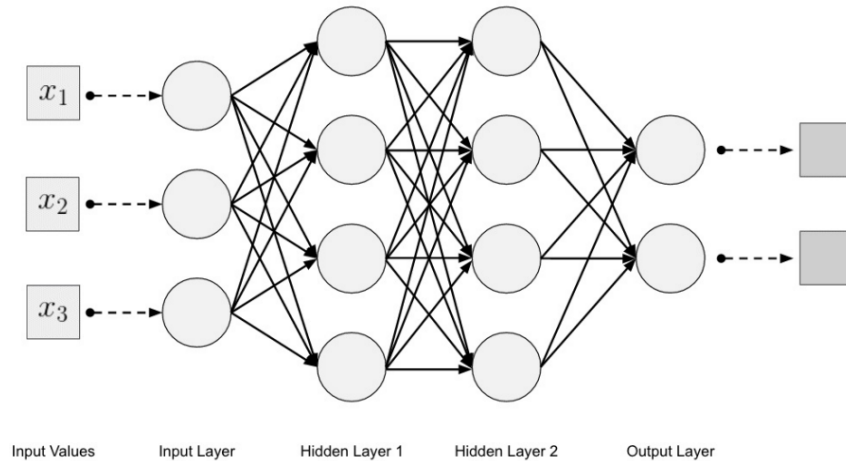


Figura 7: Esquema de Rede Neural Artificial. Fonte: [5]

Na figura 7 estão evidenciados os três grupos comumente presentes em redes neurais:

- Camada de entrada: onde os padrões são apresentados à rede;
- Camadas intermediárias ou escondidas: onde a maior parte do processamento ocorre, através das conexões ponderadas;
- Camada de saída: camada onde o resultado é apresentado.

Considerando o modelo do neurônio artificial (Figura 8), o seu funcionamento segue a seguinte lógica:

- Sinais são apresentados à entrada;
- Cada sinal é multiplicado por um peso, que indica a sua influência na saída da unidade;
- É feita a soma ponderada dos sinais que produz um nível de atividade;

- Se este nível de atividade exceder um certo limite (*threshold*) a unidade produz uma determinada resposta de saída.

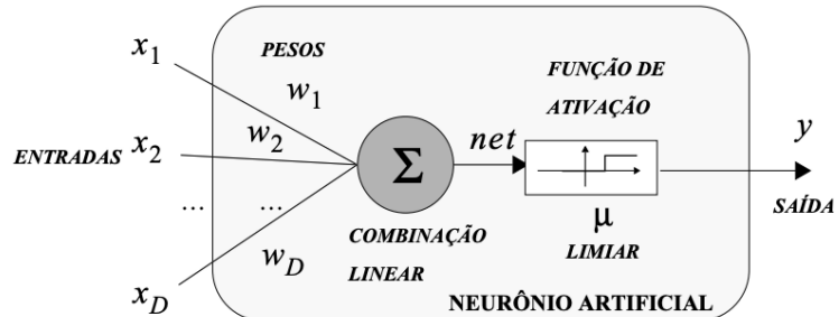


Figura 8: Modelo do Neurônio Artificial.

Suponha que tenhamos D sinais de entrada X_1, X_2, \dots, X_D e pesos w_1, w_2, \dots, w_D e limitador t ; com sinais assumindo valores booleanos (0 ou 1) e pesos valores reais.

Neste modelo, o nível de atividade a é dado por:

$$a = w_1 \cdot X_1 + w_2 \cdot X_2 + \dots + w_D \cdot X_D$$

Assim, a saída y é dada por:

- $y = 1$, se $a \geq t$ ou
- $y = 0$, se $a < t$.

Uma rede neural é especificada pela sua topologia, pelas características dos nós e pelas regras de treinamento. A seguir, será explicada a lógica do aprendizado através do algoritmo de retropropagação.

O processo de aprendizado acontece através de iterações de ajustes aplicados aos seus pesos. Através da retropropagação, o modelo descobre os pesos corretos que devem ser aplicados aos nós de uma rede neural, comparando as saídas atuais da rede com as saídas desejadas ou as corretas. A diferença entre a saída desejada e a saída atual é calculada pela função de Perda, ou Custo. Ou seja, a função de Perda informa o nível de precisão da rede neural fazendo previsões para uma determinada entrada.

Em geral, os pesos e desvios são definidos inicialmente com valores aleatórios, que com frequência produzem um valor alto de Perda quando inicia-se o treinamento de uma rede neural.

Com o treinamento, o algoritmo ajusta cada peso para minimizar a diferença entre o valor calculado e o valor correto. O termo "retropropagação" vem do fato de que o algoritmo volta e ajusta os pesos e desvios após calcular uma resposta. Quanto menor é a Perda de uma rede, mais precisa ela se torna. Logo, o processo de aprendizado pode ser determinado como a redução da saída da função de Perda [24].

2.4.1 Redes Neurais Convolucionais

As Redes Neurais Convolucionais são um tipo de redes neurais artificiais *feedforward* que realizam operações com dados de topologia matricial, tendo inúmeras aplicações em visão computacional [6].

A principal diferença deste tipo de rede neural para as demais é o fato de se utilizar da operação de convolução, em que ocorre uma multiplicação de cada elemento da matriz pelo filtro, ou *kernel*, da convolução, sendo a saída da operação o mapa de *features* extraídas pela rede. A operação de convolução é tipicamente descrita por um asterisco:

$$s(i, j) = (I * K)(i, j) = \sum_m \sum_n K(m, n) I(i + m, j + n) \quad (2.1)$$

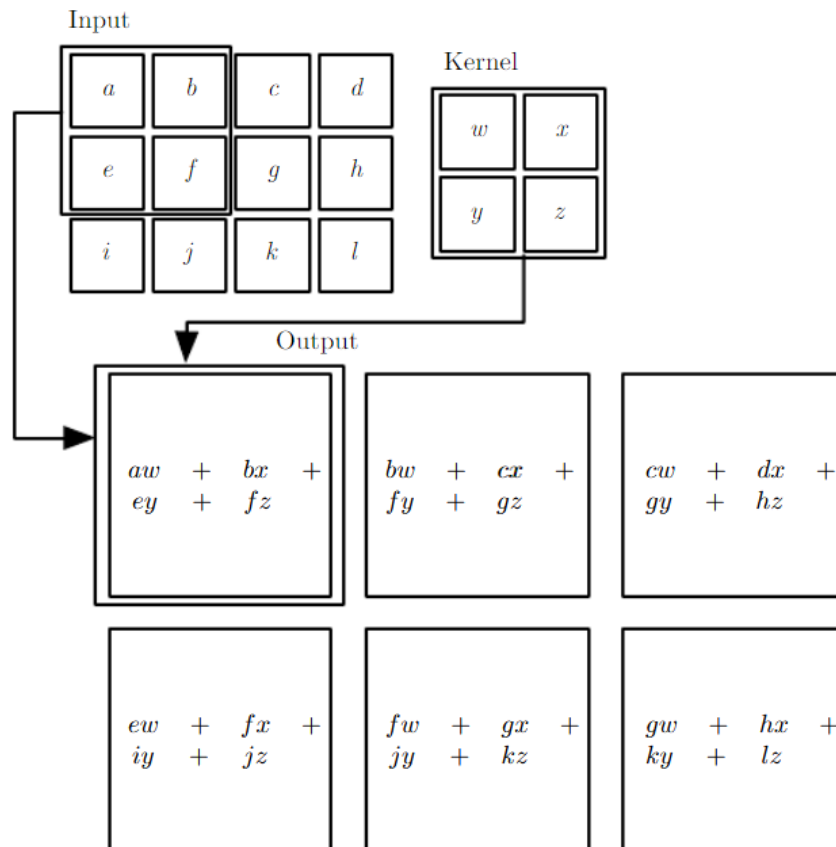


Figura 9: Exemplo de convolução 2D. A saída foi restringida a somente posições em que o *kernel* é totalmente preenchido pela imagem, chamada de convolução "válida". Fonte: [6]

Uma típica camada de rede convolucional é composta por três etapas: na primeira, são realizadas as convoluções em paralelo para produzir uma série de ativações lineares; na segunda etapa, se passa cada ativação uma função de ativação não linear, como a função ReLU (*rectified linear unit*); na terceira etapa, é utilizada uma função de *pooling* para modificar a saída da camada [6].

A função de *pooling* substitui um ponto da saída em uma determinada posição por uma síntese estatística das saídas próximas ao ponto. Isso é feito para que a representação não varie muito com pequenas translações da entrada. Ou seja, pequenas movimentações na entrada não interferem fortemente na saída. A função *max pooling*, por exemplo, reporta o valor máximo dentro da região retangular em que o ponto está:

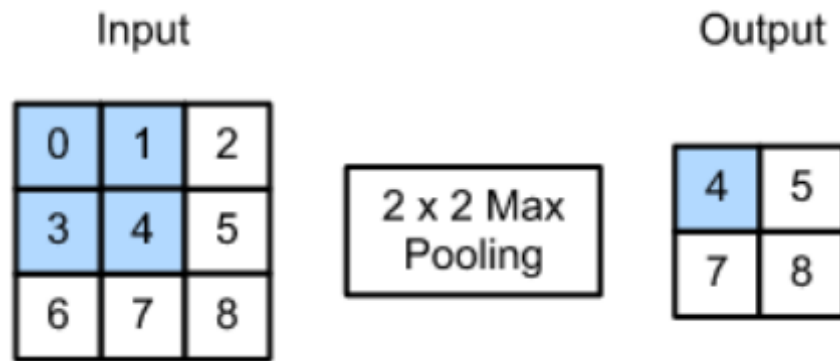


Figura 10: Exemplo de aplicação de *max pooling* de dimensão 2x2. Fonte: [7]

As camadas de convolução são responsáveis pela extração de *features* das imagens, sendo que, quanto mais camadas, maior a capacidade da rede de identificar características. No entanto, é necessário, também, um grande volume de imagens para treinar a rede.

2.4.2 VGG-16

A rede VGG-16 surgiu no contexto da competição ImageNet Large Scale Visual Recognition Challenge (ILSVRC), a competição consistia em 2 frentes de atuação. A primeira delas consistia em detectar objetos dentro de uma imagem provenientes de 200 classes. Já, a segunda consistia em classificar imagens, dentre 1000 categorias. A rede proposta por Karen Simonyan e Andrew Zisserman do *Visual Geometry Group Lab of Oxford University* [25] ficou em primeiro e segundo lugar nas duas categorias respectivamente. Atingindo acurácia de 92.7% de precisão no conjunto de dados ImageNet, que contém 14 milhões de imagens pertencentes a 1000 classes.

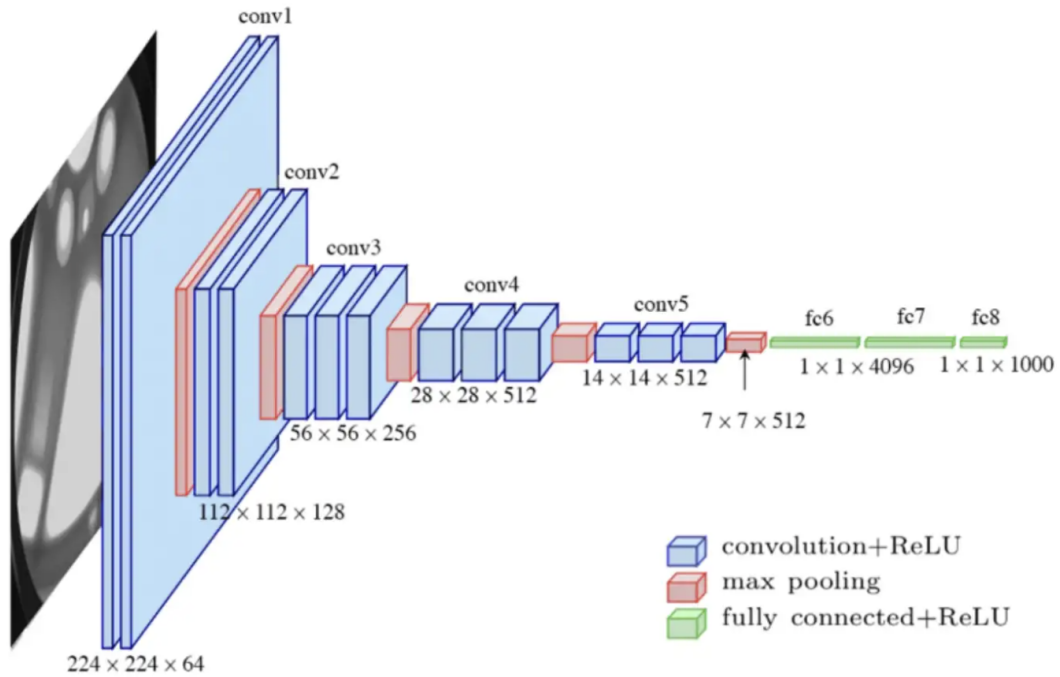


Figura 11: A arquitetura da VGG-16

A VGG-16 é uma rede neural convolucional que tem como principal atributo o aumento da profundidade da rede usando uma arquitetura com filtros convolucionais muito pequenos (3x3). Na VGG-16 existem treze camadas convolucionais, cinco camadas Max Pooling e três camadas densas, totalizando 21 camadas, porém apenas 16 com parâmetros apreensíveis.

Esta configuração mostrou uma melhora significativa perante as demais configurações da ConvNet. Por outro lado a rede possui muitos parâmetros de peso o que afeta seu desempenho e treinamento de forma significativa.

2.5 Data Augmentation

Data Augmentation é um conjunto de técnicas usadas para lidar com dados limitados que são um grande inconveniente na aplicação de modelos de aprendizagem profunda. Outro obstáculo que pode ser contornado com algumas dessas técnicas é o *overfitting*, que ocorre quando o modelo mostra-se adequado apenas para os dados de treino, como se

tivesse apenas decorado os dados de treino e não fosse capaz de generalizar para outros dados nunca vistos antes, tendo, assim, um desempenho ruim.

As primeiras demonstrações que mostram a eficácia das técnicas de Data Augmentation vêm de transformações simples como:

1. **Flip:** Flipping é uma técnica em que é feita uma inversão na imagem original, onde essa inversão pode ser na horizontal ou na vertical;
2. **Rotation:** Os aumentos de rotação são feitos girando a imagem para a direita ou esquerda em um eixo entre 10° e 359° ;
3. **Scale:** Com a técnica Scale ou Zoom, a imagem pode ser dimensionada com uma escala maior ou menor. Ao dimensionar com uma escala maior, o tamanho da imagem final será maior que o tamanho da imagem original. A maioria dos frameworks de imagens recorta uma seção da nova imagem, com tamanho igual ao da imagem original;
4. **Crop:** Ao contrário da técnica de scale, apenas amostramos aleatoriamente uma seção da imagem original. Em seguida, redimensionamos esta seção para o tamanho da imagem original;
5. **Translation:** Mudar as imagens para a esquerda, direita, para cima ou para baixo pode ser uma transformação muito útil para evitar distorções posicionais nos dados;

No entanto, essas imagens transformadas têm intrinsecamente uma distribuição semelhante às originais, levando a uma melhora de desempenho limitada.

2.5.1 Generative Adversarial Networks

As Redes Generativas Adversárias (Figura 12) são redes neurais profundas que tem como objetivo aproximar uma distribuição de dados $x \sim p_{\text{data}}$. Esse objetivo é alcançado otimizando em conjunto duas redes adversárias: um gerador e um discriminador. O gerador G, que pode ser pensado como análogo a uma quadrilha de falsificadores de notas,

é treinado para sintetizar uma imagem que se aproxime da verdadeira distribuição dos dados a partir de um vetor de ruído. Já o discriminador D , análogo à polícia, tentando detectar a moeda falsa, é otimizado para distinguir com precisão entre as imagens sintetizadas produzidas pelo gerador e as imagens reais da distribuição de dados [26]. A introdução das GANs por Goodfellow *et al.* (2014) [27] levou a modelagem generativa e a síntese de dados a níveis de qualidade nunca antes alcançados [28].

Em sua formulação mais básica, as GANs são treinados para otimizar a seguinte função valor do jogo minimax entre G e D [27]:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [1 - \log D(G(z))] \quad (2.2)$$

Aqui a rede geradora $G(z)$ com parâmetros θ_G é alimentada com uma variável aleatória $z \sim p_z(z)$ amostrada de uma dada distribuição a priori que G tenta mapear para $x \sim p_{\text{data}}$. Para conseguir isso, a rede discriminadora D com parâmetros θ_D é treinada para diferenciar entre amostras reais $x \sim p_{\text{data}}$ de um determinado *dataset* e amostras falsas $\hat{x} \sim p_{\theta_G}(x | z)$ produzidas pelo gerador. Ao fazer isso, o gerador é forçado a produzir amostras cada vez mais realistas com o objetivo de fazer com que o discriminador as classifique erroneamente como reais.

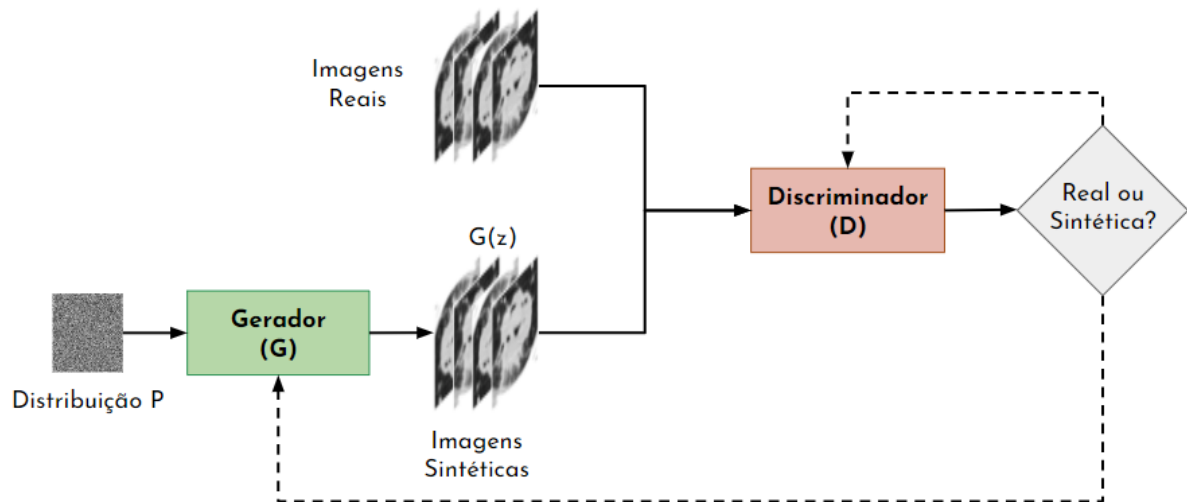


Figura 12: Arquitetura da GAN: O objetivo do gerador (G) é gerar imagens que parecem o mais realistas possível, na tentativa de enganar o discriminador (D), um classificador que tenta classificar as imagens geradas como reais ou falsas. O objetivo do treinamento é alcançado quando D classifica as imagens geradas com 50% de confiança. Fonte: [8]

O treinamento de GANs não é trivial. Como o aprendizado envolve duas redes adversárias, as GANs são conhecidas por sofrerem de vários problemas de treinamento, dentre eles o mais difícil de resolver é o *mode collapse*. O *mode collapse* ocorre quando o gerador fica parado, produzindo apenas um (ou alguns) modos da distribuição de dados de entrada. Isso leva a uma perda de diversidade nos conjuntos de dados gerados que podem prejudicar muito o desempenho de redes treinadas com os dados gerados [28].

2.5.2 Progressive Growing of GANs

Buscando resolver alguns dos problemas comuns no treinamento das GANs Karras *et al.* [29] propuseram a PGGAN (do Inglês *Progressive Growing of GANs* - FAIR) (Figura 13), uma metodologia de treinamento para GANs que com imagens de baixa resolução e, em seguida, aumenta progressivamente a resolução adicionando camadas à rede. A natureza incremental dessa metodologia de treinamento consiste em primeiro descobrir a estrutura em grande escala da distribuição da imagem e depois focar a atenção para detalhes de escala cada vez mais finos, em vez de ter que aprender todas as escalas

simultaneamente.

As redes do gerador (G) e do discriminador (D) são ampliadas por um fator de dois em estágios de treinamento consecutivos. Para começar, a saída da imagem é 4x4, depois 8x8, 16x16 e assim por diante, até 512x512, em oito estágios. Alterar escalas de imagem requer a introdução de novos blocos convolucionais na rede, mas a introdução direta leva à instabilidade do treinamento. Por essa razão, desenvolveu um novo parâmetro, alpha, para introduzir gradualmente (*fade-in*) novos blocos convolucionais pegando duas saídas do bloco anterior e conectando-as em paralelo com o novo bloco. Alpha começa com um valor de 0 e aumenta para 1 à medida que o treinamento avança. O incremento lento de alfa aumenta gradualmente a força da conexão entre os blocos de convolução anterior e novo, ao mesmo tempo em que enfraquece a conexão entre o bloco de convolução anterior e a saída. Outros conceitos como taxa de aprendizado equalizada e normalização pixel a pixel também são aplicados no PGGAN, contribuindo para melhorar a estabilidade e geração de imagens realistas de alta qualidade [8].

Na tentativa de reduzir o efeito do *mode collapse* e estabilizar o procedimento de aprendizado, ambos problemas comuns nas GANs, a PGGAN utiliza a Distância de Wasserstein (Equação 2.3) como função de perda:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|] \quad (2.3)$$

Onde $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ denota o conjunto de todas as distribuições conjuntas $\gamma(x, y)$ cujas distribuições marginais são respectivamente \mathbb{P}_r e \mathbb{P}_g [30]. As Wasserstein GANs (WGANs), um tipo de GAN que utiliza essa métrica como função de perda, são consideradas como um dos principais avanços para superar os desafios de treinamento de GANs [28].

Buscando melhorar ainda mais o desempenho das GANs Gulrajani *et al.* [31] propuseram o uso da perda de Wasserstein (Equação 2.3) juntamente com penalidade de gradiente (WGAN-GP). Os resultados do trabalho demonstraram treinamento estável de arquiteturas variadas GANs, melhorias de desempenho em relação a WGAN e geração de

imagens de alta qualidade. A Equação 2.4 representa a função de perda da WGAN-GP:

$$L = \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [D(\tilde{x})] - \mathbb{E}_{x \sim \mathbb{P}_r} [D(x)] + \underbrace{\lambda \mathbb{E}_{\tilde{x} \sim \mathbb{P}_{\tilde{x}}} [(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]}_{\text{penalidade de gradiente}} \quad (2.4)$$

Onde \mathbb{P}_r é a distribuição de dados e \mathbb{P}_g é a distribuição do modelo implicitamente definida por $\tilde{x} = G(z)$ sendo $z \sim p(z)$ a entrada (z) para o gerador que é amostrada de alguma distribuição de ruído, como a distribuição uniforme.

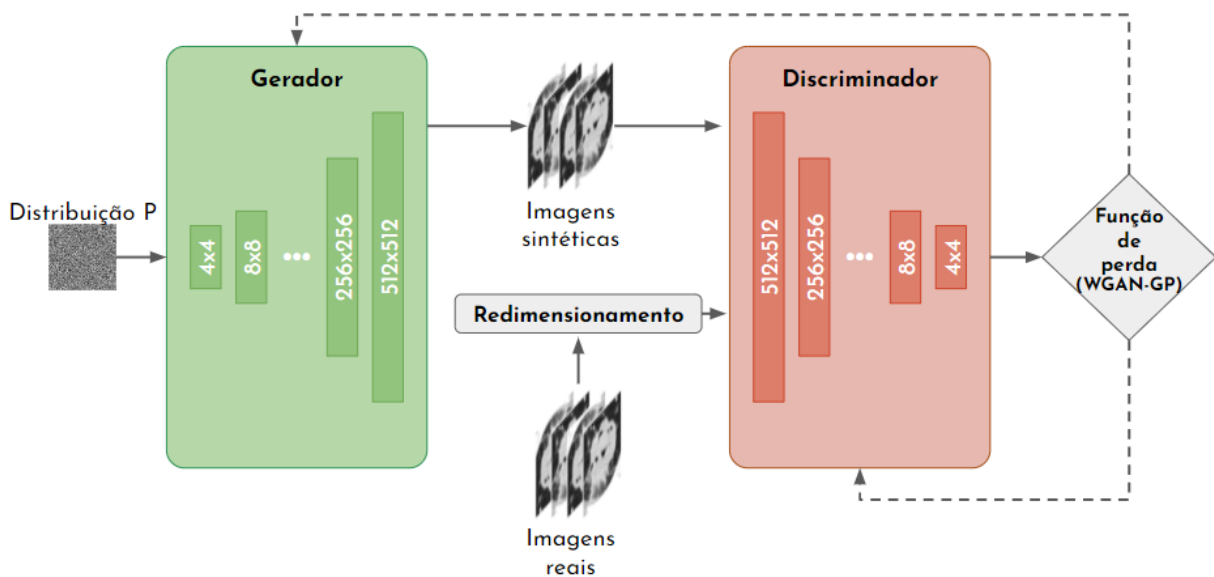


Figura 13: Arquitetura da PGGAN: No início, ambos G e D começam com uma dimensão de 4x4 e são treinados até a convergência. Na etapa seguinte, o treinamento é retomado com imagens com a dimensão duas vezes maior, 8x8, redimensionadas por um bloco de convolução. O processo se repete e as imagens aumentam gradualmente em escala por potências de 2 até que o tamanho necessário das imagens sintéticas seja alcançado (512x512). Fonte: [8]

No contexto do uso da PGGAN para a geração de imagens de treinamento adicionais para classificadores, Han *et al.* [20] mostraram que as PGGANs podem gerar imagens realistas e obter maior desempenho na detecção de tumores. Isso ocorre porque a geração de imagens por meio de PGGANs obtém boa generalização e sintetiza imagens com a distribuição não preenchida pelo conjunto de dados original, o que aumenta a variabilidade dos dados de treinamento e conseqüentemente a habilidade de generalização do classificador.

3 TECNOLOGIAS UTILIZADAS

Esta seção descreve as principais tecnologias utilizadas no projeto e o motivo para o qual foram utilizadas durante o desenvolvimento do projeto.

3.1 Python

A linguagem Python se tornou o padrão para o aprendizado de máquina [32]. Grande parte da sua popularidade com machine learning é devido a sua simplicidade e a grande quantidade de bibliotecas para todos os aspectos do aprendizado de máquina aplicado. Além disso, uma das principais vantagens de usar o Python é a capacidade de interagir diretamente com o código, usando um terminal ou outras ferramentas como o Jupyter Notebook [33].

3.1.1 Jupyter Notebook

O Jupyter Notebook é um ambiente interativo para execução de código no navegador. É uma ferramenta para análise exploratória de dados e é amplamente utilizada por cientistas de dados, facilitando a incorporação de código, texto e imagens [33].

3.1.2 NumPy

O NumPy é uma ferramenta básica para a computação científica. Ele permite, entre outras coisas: a manipulação e o armazenamento de dados de *arrays* multidimensionais de maneira rápida e eficiente, a realização de cálculos elementares e operações matemáticas

vetoriais, leitura e gravação de conjuntos de dados baseados em arrays em disco, operações de álgebra linear e geração de números aleatórios [34].

3.1.3 Matplotlib

A fim de criar visualizações usamos a biblioteca Matplotlib, utilizada principalmente para exibir as imagens durante os processos de análise exploratória e treinamento [35].

3.2 PyTorch

O framework escolhido para o desenvolvimento da solução de *machine learning* é o PyTorch, um projeto *open-source* cujo principal desenvolvedor é o laboratório de Pesquisa em AI do Facebook (do Inglês *Facebook's AI Research* - FAIR), devido à facilidade de utilização, à flexibilidade, aos recursos de *debugging*, a curta duração de treinamento e por ser a principal biblioteca usada em pesquisas acadêmicas nos últimos anos [36].

O *framework* conta com os tensores PyTorch, *arrays* multidimensionais usadas como base para todas as operações avançadas do *framework*. Os tensores são semelhantes aos *ndarrays* do NumPy, exceto que os tensores podem ser executados em GPUs para acelerar a computação [37].

3.3 Hardware

Tanto o pré-processamento das imagens quanto o treinamento e a validação do modelo serão feitos em um ambiente virtual que utiliza uma unidade de processamento gráfico (do Inglês *graphics processing unit* - GPU) NVIDIA Tesla T4 [38] com 16Gb de memória.

4 MATERIAIS E MÉTODOS

4.1 Datasets Utilizados

Com o objetivo de ter aumentado o número de imagens e a variabilidade das mesmas foram utilizados três *datasets* diferentes nesse estudo: HCFMRP (Seção 4.1.1), Lung-PET-CT-Dx (Seção 4.1.2) e NSCLC Radiomics (Seção 4.1.3). Juntos os *datasets* contêm 3.263 imagens, de 205 exames, cujo diagnóstico foi adenocarcinoma e 4.274 imagens, de 220 exames, cujo diagnóstico foi carcinoma escamoso. A Tabela 2 agrega as informações sobre as imagens contidas em cada um dos *dataset* utilizados.

Dataset	Número de exames		Número de fatias contendo nódulo		Espessura mediana das fatias (mm)	
	ADC	SCC	ADC	SCC	ADC	SCC
HCFMRP	63	30	1459	863	1,0	1,0
Lung-PET-CT-Dx	91	40	894	793	2,0	2,0
NSCLC	51	150	910	2618	3,0	3,0

Tabela 2: Informações sobre as imagens contidas nos datasets utilizados

4.1.1 HCFMRP

O *dataset* usado neste trabalho é composto por 205 imagens de TC de pacientes cujo câncer de pulmão foi confirmado histologicamente ou cirurgicamente. Os pacientes foram diagnosticados e tratados no Hospital das Clínicas da Faculdade de Medicina de Ribeirão Preto (HCFMRP) ou no Hospital Estadual de Américo Brasiliense durante os anos de 2010 a 2017 [4].

As imagens foram reconstruídas utilizando aquisições volumétricas com matriz de 512×512 pixels, espessura de corte de 0,5–1,25 mm e espaçamento de pixel de 0,58–1,07 mm.

O formato de arquivo das imagens é o padrão de comunicação de imagens digitais em medicina (do Inglês *Digital Imaging and Communications in Medicine* - DICOM), um formato de arquivo projetado para visualização científica e processamento de imagens envolvendo dados multidimensionais [39]. As imagens foram semiautomaticamente segmentadas pelo algoritmo de crescimento de região volumétrica GrowCut [40] da plataforma de análise e visualização de imagens 3D-Slicer v4.3.1 [4].

Para realizar a segmentação semiautomática das imagens no 3D-Slicer, duas regiões foram marcadas, uma interna e uma externa ao tumor. Em seguida, o algoritmo GrowCut detectou tridimensionalmente o tecido tumoral. Após isto, a região externa ao tumor foi removida, e a sua borda delimitada (Figura 14(b)). A segmentação da base foi feita por José Raniery Ferreira Junior, sob supervisão do Prof. Dr. Marcel Koenigkan Santos, médico radiologista especializado em tórax [4].

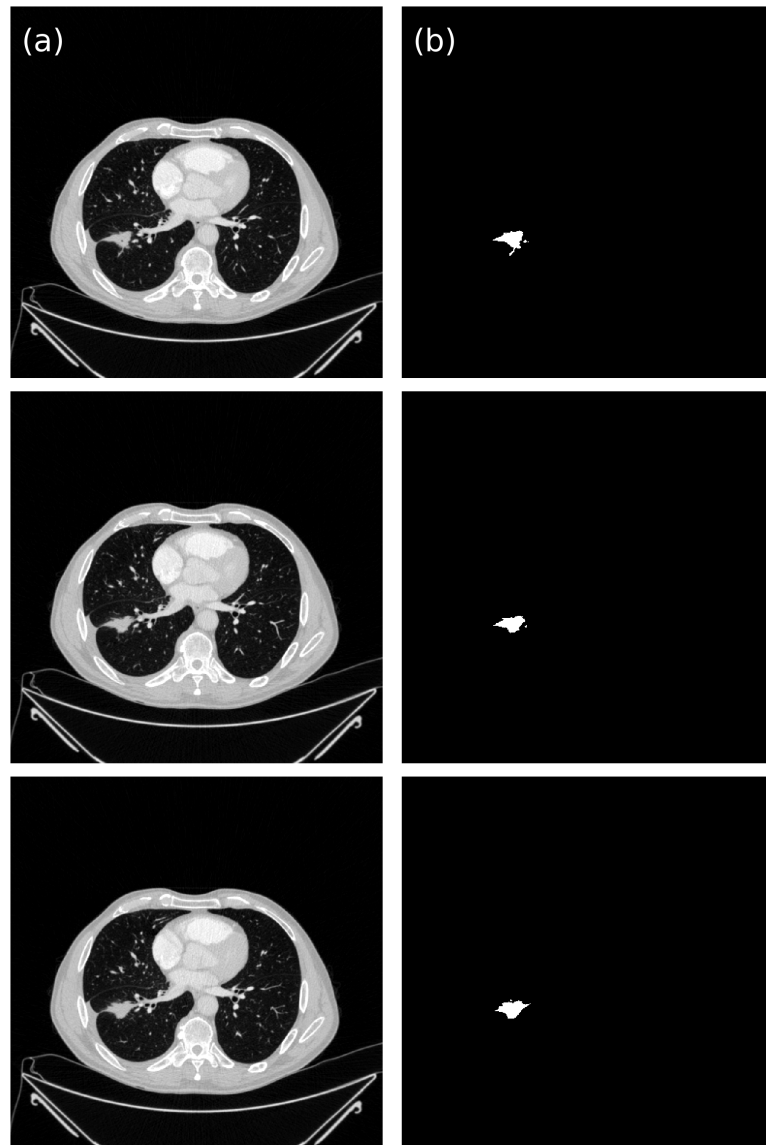


Figura 14: Exemplo de segmentação de nódulos pulmonares do *dataset* HCFMRP: Imagens sequenciais de um paciente com (a) imagem de TC e (b) máscara de segmentação correspondente lado a lado. Fonte: [4]

4.1.2 Lung-PET-CT-Dx

Este *dataset* consiste em imagens no formato DICOM de indivíduos com câncer de pulmão e arquivos de anotação no formato XML (do Inglês *Extensible Markup Language*) que indicam a localização do tumor com caixas delimitadoras. As imagens foram adquiridas retrospectivamente de pacientes com suspeita de câncer de pulmão e submetidos a biópsia pulmonar padrão e PET/CT. Os indivíduos foram classificados de acordo com o

tipo histopatológico detectado por meio de diagnóstico do tecido canceroso e a localização de cada tumor foi anotada por cinco radiologistas torácicos acadêmicos com experiência em câncer de pulmão.

Cada exame do *dataset* contém um volume de TC com resolução de 512×512 , espessura de corte de 2 mm e espaçamento de pixel de 0.54-0,90 mm. A emissão tridimensional (3D) e a varredura de transmissão foram adquiridas da base do crânio até a metade do fêmur [41] [42].

4.1.3 NSCLC Radiomics

Este conjunto de dados contém imagens de 422 pacientes com câncer de pulmão de células não pequenas (NSCLC). Para esses pacientes, estão disponíveis tomografias computadorizadas pré-tratamento, delineamento manual por um oncologista de radiação do volume 3D do volume tumoral bruto e dados de resultados clínicos (tipo/subtipo histológico dos nódulos, idade, sexo, etc). O delineamento do tumor primário feito por um oncologista está disponível nos arquivos DICOM Radiotherapy Structure Sets (RTSTRUCT) e DICOM Segmentation (SEG) presentes em cada exame [42] [43] [44].

Cada exame do *dataset* contém um volume de TC com resolução de 512×512 , espessura de corte de 3 mm e espaçamento de pixel de 0.97 mm.

4.2 Pré-processamento das Imagens

Etapas de pré-processamento das imagens:

1. **Unificação dos datasets:** A primeira etapa foi realizar o agrupamento das imagens oriundas dos três *datasets* diferentes descritos na Seção 4.1 em um único *dataset*.
2. **Conversão para HU:** Os arquivos de imagem dos exames em formato DICOM foram lidos, convertidos para *arrays* do NumPy e os valores das unidades de Hounsfield de cada imagem foram calculados a partir dos valores de Rescale Intercept e

Rescale Slope disponíveis nos arquivos DICOM.

3. **Seleção das fatias com nódulos:** As imagens volumétricas dos exames tem número variável de cortes e dado que para um mesmo exame nem todos os cortes contem um nódulo, a primeira etapa de processamento foi selecionar apenas os cortes que continham um nódulo, isto foi feito usando-se as mascaras de segmentação para os *datasets* NSCLC Radiomics e HCFMRP (Figura 14(b)) e as caixas delimitadoras para o *dataset* Lung-PET-CT-Dx.
4. **Extração da ROI:** Nessa etapa se deu a extração de um região de interesse (do Inglês *region of interest* - ROI) retangular contendo um nódulo pulmonar, a região retangular foi estendida em cada lado por 5 pixels, para incluir também os arredores do nódulo (Figura 15(b)), dado que a informação contida nas vizinhanças do tumor é rica e pode auxiliar na classificação, especialmente, em tumores pequenos [45]. A etapa de extração da ROI melhorara a precisão da classificação, pois o restante do tecido pulmonar que deixa de ser usado pode afetar os resultados. Ao mesmo tempo, isso reduz o tempo de treinamento porque o modelo precisa se concentrar em regiões menores e permite que sejam usados tamanhos de mini-batch maiores [46].
5. **Redimensionamento:** Em seguida, todas as imagens foram redimensionados para o tamanho 64 x 64 pixels e normalizadas para um intervalo de valores entre 0 e 1 (Figura 15(c)), pois isso ajuda a reduzir a influência de outros redimensionamentos e aumenta a capacidade das redes neurais de capturar características nas imagens [47].
6. **Conversão para tensores PyTorch:** Por último as imagens são convertidas para tensores PyTorch (Seção 3.2) para que possam ser usadas para treinar e validar tanto o modelo de classificação (VGG16) quanto o modelo de geração de imagens sintéticas (GAN).

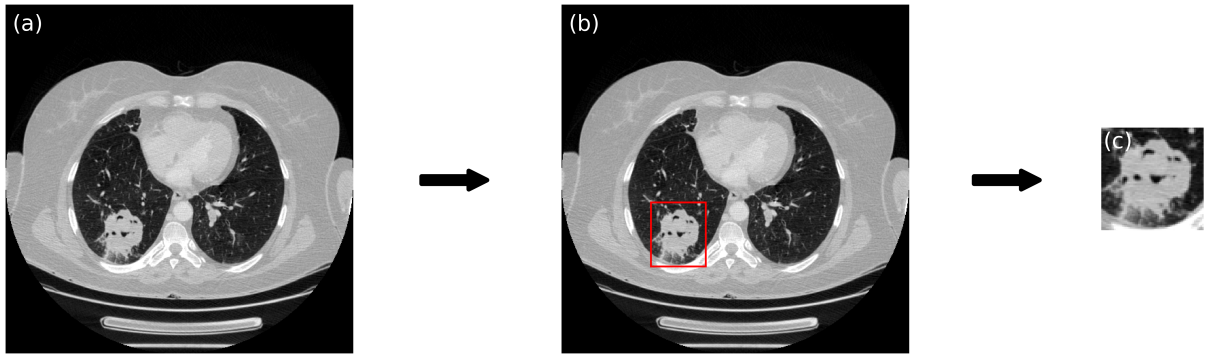


Figura 15: Pré-processamento das imagens: (a) imagem de TC com tamanho 512 x 512, (b) ROI retangular centrada no nódulo com tamanho 94 x 81, (c) ROI normalizada e redimensionada com tamanho 64 x 64. Fonte: Autor

5 PROJETO E IMPLEMENTAÇÃO

5.1 Progressive Growing GAN

5.1.1 Detalhes de implementação

A rede neural foi implementada seguindo a arquitetura original da PGGAN [29] com a Loss de Wasserstein (Equação 2.3) usando penalidade de gradiente. A taxa de aprendizado foi de $1,0 \times 10^3$ para o otimizador Adam [48]. Devido a restrições de memória o tamanho de lote variou de acordo com a resolução das imagens, com os maiores tamanhos para as menores resoluções e vice-versa.

5.1.2 Treinamento da rede

Foram treinadas PGGANs diferentes para cada tipo histológico de câncer. Diferentemente da proposta original descrita na Seção 2.5.2 a PGGAN desse trabalho foi treinada apenas até a resolução 64 x 64 pixels. Os tamanhos de lote variavam, de acordo com o resolução da imagens, começando com 64 para imagens 4x4 pixels e diminuindo para 32 para imagens 64x64 pixels, por exemplo. Foi utilizado o otimizador Adam e uma taxa de aprendizado de 0,001. β_1 e β_2 foram ajustados para 0 e 0,99, respectivamente.

A quantidade de iterações antes do treinamento ser interrompido variou de 15000 a 37500 e tempo de treinamento variou de 6 a 9 horas, ambos a depender do tipo histológico das imagens a serem geradas e dos hiperparametros selecionados.

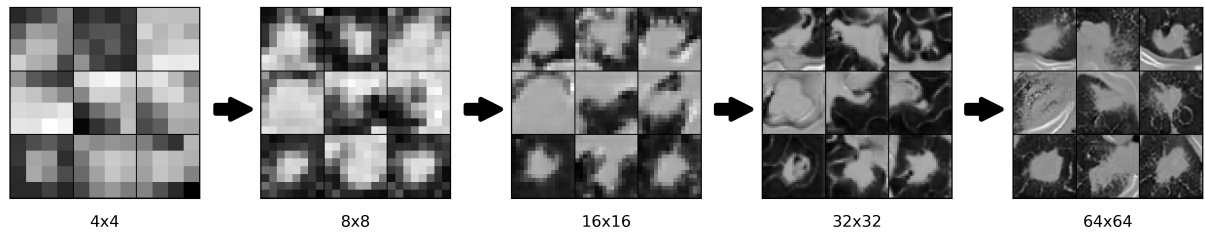


Figura 16: Exemplo de imagens sintetizadas pelo gerador em diferentes resoluções ao longo do treinamento da PGGAN, mostrando o aprendizado progressivo da rede. Fonte: Autor

5.2 VGG-16

5.2.1 Detalhes de implementação

A rede neural foi implementada com base a arquitetura original da VGG16 [25], com a diferença que a dimensão da camada de input foi alterada de 244 x 244 pixels para 64 x 64 pixels. A função objetivo utilizada foi a Regressão Logística Multinomial [49]. O otimizador utilizado foi o Gradiente Descendente Mini-Batch (do Inglês *Mini Batch Gradient Descent*) com momento de 0,9 e tamanho do lote de 32. O treinamento foi regularizado por decaimento de peso (com multiplicador de penalidade L2, termo extra adicionado à função de custo com a intenção de reduzir o *overfitting*, definido para 5×10^{-4}) e regularização por *dropout* [50] para as duas primeiras camadas *fully-connected* (com taxa *dropout* de 0,5). A taxa de aprendizado foi definida para 10^{-3} .

5.2.2 Treinamento da rede

Para o treinamento da VGG16 usando apenas imagens reais, o *dataset* unificado foi dividido de forma aleatória em três outros *datasets*: treino, teste e validação. A composição desses *datasets* em relação ao número de imagens e ao tipo histológico dos nódulos é a seguinte:

- Conjunto de treinamento: 2397 imagens de ADC e 3011 imagens de SCC.
- Conjunto de validação: 299 imagens de ADC e 373 imagens de SCC.

- Conjunto de teste: 264 imagens de ADC e 408 imagens de SCC.

O treinamento foi interrompido após 8500 iterações (50 *epochs*), quando a acurácia de validação parou de aumentar. O tempo de treinamento foi de 14 minutos e 30 segundos.

Já quando aplicou-se *data augmentation* com as imagens geradas pela PGGAN, as imagens sintéticas foram adicionadas apenas ao *dataset* de treino para manter a imparcialidade da avaliação.

6 AVALIAÇÃO

6.1 Avaliação da etapa de geração de imagens sintéticas

No que diz respeito a avaliação de imagens sintéticas geradas por GANs em alguns estudos [20] um médico especialista avalia quantitativamente quão realistas são as imagens geradas. O chamado Teste de Turing Virtual (do Inglês *Visual Turing Test*) [51] é usado para sondar a capacidade humana de identificar atributos e relacionamentos em imagens, e também na avaliação da qualidade visual das imagens geradas por GAN. Entretanto, como não havia um médico especialista disponível para avaliar as imagens neste trabalho foi preciso recorrer a técnicas matemáticas.

No entanto, mesmo usando técnicas matemáticas a avaliação e comparação de GANs continua sendo uma tarefa desafiadora [10]. As GANs ganharam atenção significativa nos últimos anos, com várias variantes propostas e utilizadas em muitas aplicações e embora várias métricas tenham sido introduzidas, ainda não há consenso sobre qual métrica captura melhor os pontos fortes e as limitações dos modelos e deve ser usada para comparação justa dos modelos [26].

Todavia, apesar de não haver consenso sobre qual a melhor métrica a ser usada na avaliação algumas métricas são amplamente utilizadas por possuírem propriedades desejáveis para uma métrica de avaliação de GANs eficiente, como por exemplo, favorecer modelos que geram amostras fidedignas (ou seja, capacidade de distinguir amostras geradas das reais), favorecer modelos que geram amostras com alta variabilidade e concordar com julgamentos e classificações humanas dos modelos [26].

Dentre essas métricas uma das mais usadas é a Frechet Inception Distance (FID) [28], outras métricas utilizadas, especialmente para medir a utilidade de GANs para gerar amostras de treinamento adicionais, caso desse estudo, são os escores GAN-train e GAN-test [10]. As métricas serão descritas com mais detalhe na Seção 6.1.1.

6.1.1 Métricas utilizadas

6.1.1.1 Fréchet Inception Distance (FID)

O Fréchet Inception Distance (FID) faz uso de uma rede Inception pré-treinada para avaliar a qualidade das imagens geradas pela GAN [28]. O FID é a distância entre a distribuição do conjunto de imagens geradas pela GAN e o conjunto de imagens reais utilizadas para treinar a GAN. Dados os dois conjuntos de dados, o FID é definido como a distância de Fréchet das distribuições das Inception *features* dos dois conjuntos [52]. As Inception *features* são as ativações da penúltima camada de uma rede Inception pré-treinada. Sendo (μ_r, Σ_r) e (μ_g, Σ_g) a média e a covariância da distribuição real e da distribuição gerada pela GAN, respectivamente, o FID é definido como:

$$FID(r, g) = \|\mu_r - \mu_g\|_2^2 + \text{Tr} \left(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{\frac{1}{2}} \right) \quad (6.1)$$

Valores pequenos de FID correspondem a GANs com melhor desempenho, pois representam menores distâncias entre as distribuições de dados sintéticos e reais.

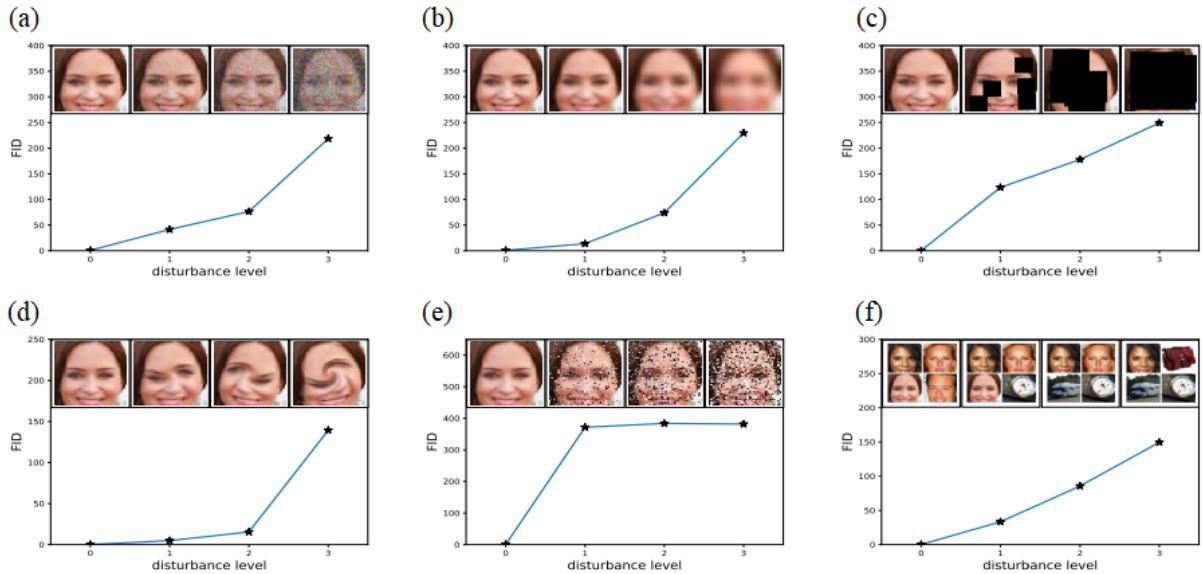


Figura 17: Exemplo da sensibilidade do FID à distorções de imagem: (a) ruído gaussiano, (b) desfoque gaussiano, (c) inserção de retângulos pretos, (d) imagens espiraladas, (e) ruído de sal e pimenta e (f) *dataset* CelebA contaminado por imagens do *dataset* ImageNet. O nível de perturbação aumenta de zero para o nível mais alto. O FID capta bem o nível de perturbação crescendo monotonamente. Fonte: [9]

Todavia, o FID conta com algumas limitações que devem ser consideradas no momento da avaliação, sendo uma delas o fato de que a comparação aproxima as ativações de imagens reais e geradas como distribuições gaussianas, o que nem sempre é verdade, e calcula as médias e covariâncias (Equação 6.1) das distribuições, que são estatísticas muito simplistas para capturar detalhes sutis em imagens. Além disso, a métrica usa uma rede Inception pré-treinada no *dataset* ImageNet, um *dataset* de cenas naturais, que está longe de ser ideal para outros conjuntos de dados, como imagens médicas [10].

6.1.1.2 GAN-train e GAN-test

No caso desse estudo há interesse em medir a capacidades das GANs gerarem amostras de treinamento adicionais, para isso as imagens geradas não devem ser apenas realista, mas também reconhecíveis como proveniente de uma determinada classe (ADC ou SCC). Para tal, Shmelkov *et al.* [10] propuseram comparar GANs com dois escores, GAN-train e GAN-test:

- **GAN-train:** Acurácia (Equação 6.2) de uma rede neural treinada em imagens geradas por GAN e avaliada em imagens reais.
- **GAN-test:** Acurácia (Equação 6.2) de uma rede neural treinada em imagens reais e avaliada em imagens geradas por GAN.

Para o GAN-train, a intuição é que em uma GAN não ideal a acurácia GAN-train será muito menor que a acurácia de validação do classificador treinado com imagens reais. Já quando a acurácia GAN-train é próxima da acurácia de validação, isso significa que as imagens da GAN são de alta qualidade e tão diversas quanto o conjunto de treinamento.

Para o GAN-test, a intuição é que em uma GAN não ideal a acurácia GAN-test será muito menor que a acurácia de validação quando a GAN não captura bem a distribuição das imagens reais e a qualidade da imagem é ruim. Já quando a acurácia GAN-test é muito maior, isso significa que a GAN está sofrendo de *overfitting* (quando o modelo tem um bom desempenho nos dados de treino, porém, não generaliza bem para novos dados) e simplesmente memoriza o conjunto de treinamento. É importante ressaltar que dado que um modelo que memoriza perfeitamente as imagens de treinamento terá uma alta pontuação no score GAN-test, essa métrica não captura a diversidade dos dados geradas, ela está relacionada com quantificação de quão próximas as imagens geradas estão de uma variedade de dados.

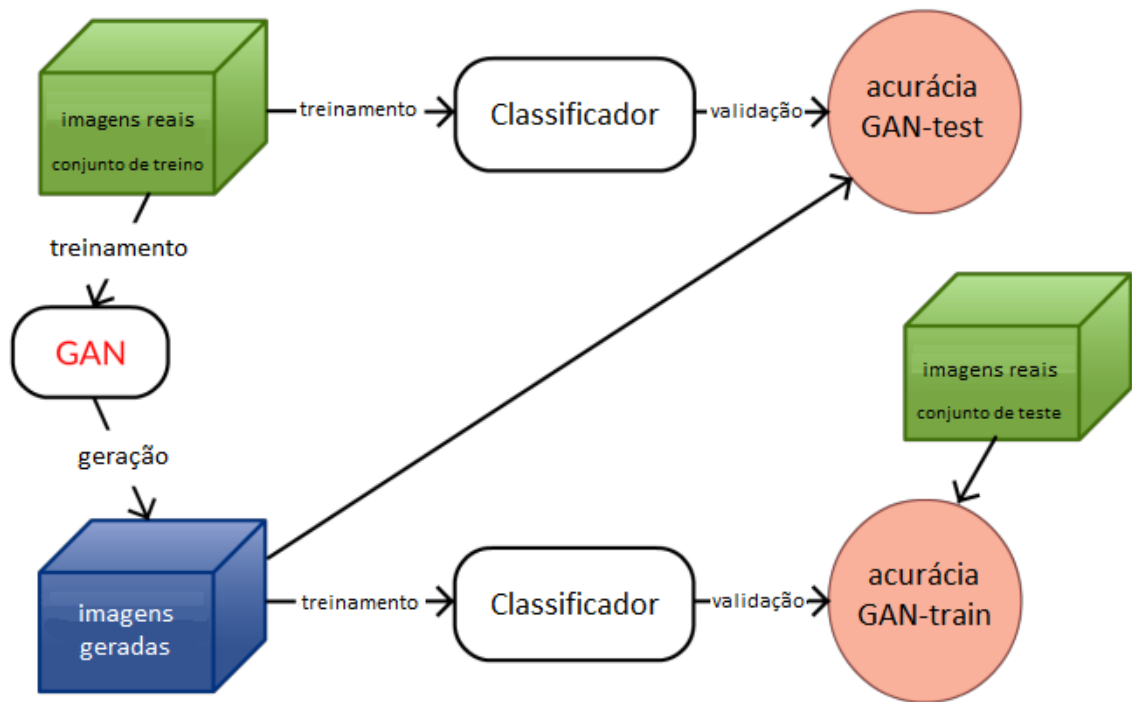


Figura 18: Ilustração do GAN-train e GAN-test: O GAN-train aprende um classificador em imagens geradas pela GAN e mede o desempenho em imagens reais. Isso avalia a diversidade e o realismo das imagens geradas. O GAN-teste aprende um classificador com imagens reais e o avalia com imagens geradas pela GAN. Isso mede o quão realistas são as imagens geradas. Fonte: [10]

No que tange às limitações as métricas mencionadas acima são indiretas e dependem fortemente da escolha do classificador usado. No entanto, eles são úteis para avaliar modelos generativos com base na noção de que um modelo generativo melhor deve resultar em melhores representações para tarefas como a classificação supervisionada) [26].

Para este estudo, a rede neural utilizada para treinar tanto o GAN-test quanto o GAN-train foi a VGG16 (Seção 2.4.2).

6.1.2 Resultados

Com o intuito de se obter a PGGAN com o melhor desempenho na geração de dados de treinamento adicionais, várias PGGANs foram treinadas com diferentes hiperparâmetros (tamanho de lote e número de *epochs*) e em seguida avaliadas de acordo com as

métricas FID, GAN-test e GAN-train. A Tabela 3 mostra os resultados da avaliação, as colunas "Número de *epochs*" e "Tamanho de lote" da Tabela 3 apresentam os valores dos hiperparâmetros para cada resolução de imagem em formato de uma lista em que os valores da esquerda para a direita correspondem às resoluções das imagens em ordem crescente de 4x4 pixels a 64x64 pixels.

Número de <i>epochs</i>	Tamanho de lote	ADC			SCC		
		FID	GAN-train	GAN-test	FID	GAN-train	GAN-test
[30,30,30,30,30]	[64,64,64,32,32]	38,348	0,528	0,606	35,486	0,528	0,606
[50,50,50,50,50]	[64,64,64,32,32]	35,048	0,652	0,609	25,179	0,777	0,609
[75,75,75,75,75]	[64,64,64,32,32]	39,163	0,674	0,527	35,856	0,677	0,527
[50,50,50,50,50]	[32,32,32,16,16]	48,605	0,659	0,509	33,724	0,756	0,509
[75,75,75,75,75]	[32,32,32,16,16]	36,580	0,610	0,491	36,398	0,683	0,491
[30,30,50,50,75]	[32,32,32,32,32]	31,852	0,753	0,618	29,356	0,695	0,618

Tabela 3: Avaliação das PGGANs treinadas com diferentes hiperparâmetros. Os valores em negrito representam o menor valor de FID ou os maiores valores dos escores GAN-test e GAN-train na coluna. Para fins comparativos, o valor da acurácia de validação para o modelo treinado e testado com imagens reais é 0,906. Fonte: Autor

Além da avaliação por meio das métricas, para se ter uma noção visual dos resultados alcançados pelas PGANs apresentamos nas Figuras 19 e 20, para cada tipo histológico, algumas imagens aleatórias geradas pelos modelos e também imagens reais para comparação.

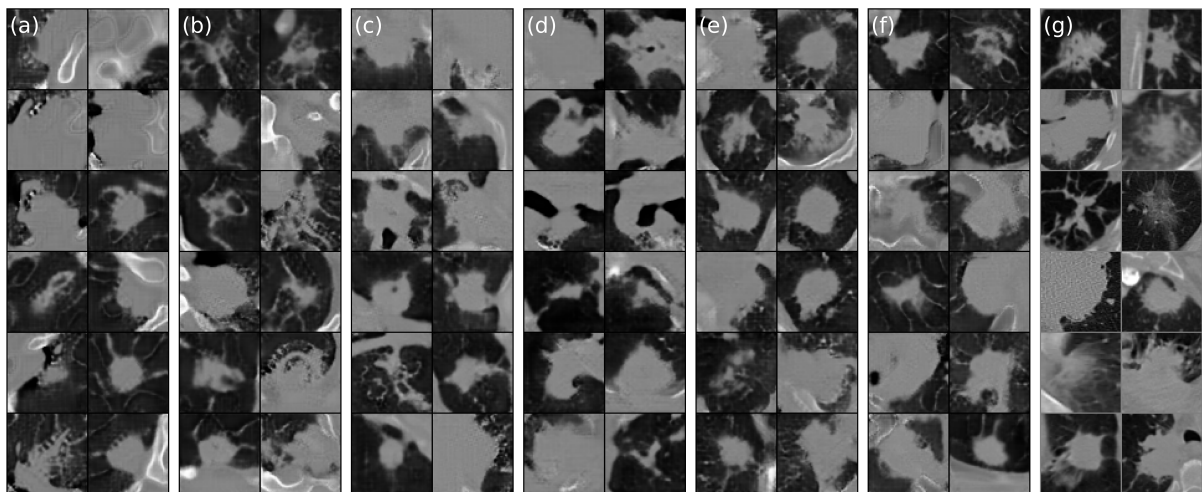


Figura 19: Comparação entre imagens geradas pelas PGGANs e imagens reais para ADC: (a), (b), (c), (d), (e) e (f) são as imagens geradas pelas PGGANs na mesma ordem em que elas são apresentadas na Tabela 3, (g) são imagens reais. Fonte: Autor

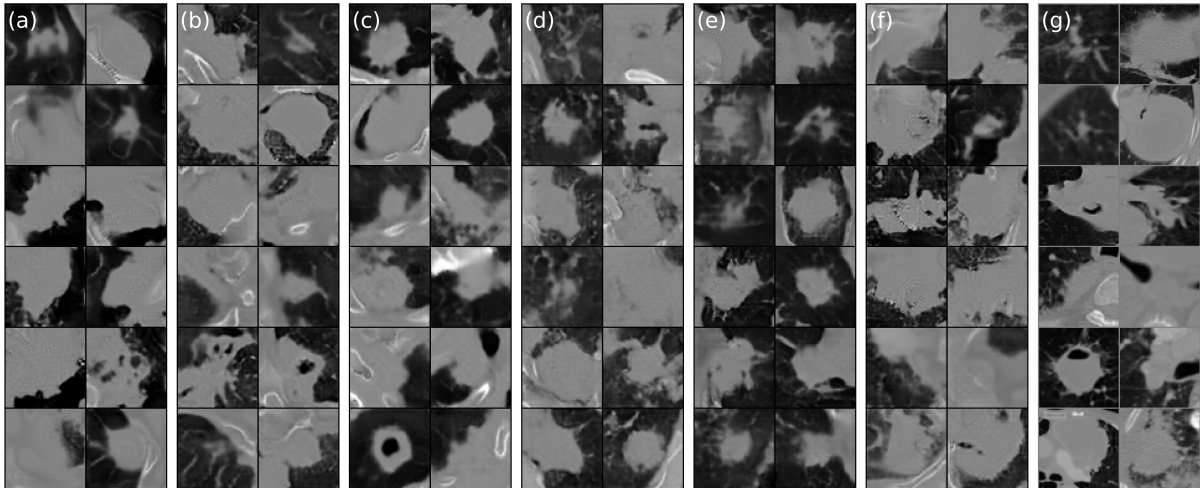


Figura 20: Comparação entre imagens geradas pelas PGGANs e imagens reais para SCC: (a), (b), (c), (d), (e) e (f) são as imagens geradas pelas PGGANs na mesma ordem em que elas são apresentadas na Tabela 3, (g) são imagens reais. Fonte: Autor

6.2 Avaliação da etapa classificação de imagens

6.2.1 Métricas utilizadas

Existem várias métricas usadas para avaliar modelos de classificação imagem [53], as mais comuns estão listadas abaixo, considerando o número de verdadeiros positivos (VP), verdadeiros negativos (VN), falsos positivos (FP) e falsos negativos (FN).

- **Acurácia:** Refere-se à porcentagem de amostras que foram corretamente classificadas (Equação 6.2).

$$Acurácia = \frac{VP + FP}{VP + VN + FP + FN} \quad (6.2)$$

- **Sensibilidade ou Recall:** refere-se à porcentagem de amostras positivas que foram corretamente classificadas pelo modelo (Equação 6.3).

$$Sensibilidade = \frac{VP}{VP + FN} \quad (6.3)$$

- **Precisão:** Refere-se à porcentagem de amostras classificadas como positivas pelo

modelo que realmente são positivas (Equação 6.4).

$$Precisão = \frac{VP}{VP + FP} \quad (6.4)$$

- **Especificidade:** Refere-se à porcentagem de amostras classificadas como negativas pelo modelo que realmente são negativas (Equação 6.5).

$$Especificidade = \frac{VN}{VN + FP} \quad (6.5)$$

- **F1-Score:** É definido como a média harmônica entre a precisão e a sensibilidade e muito usado visto que normalmente é interessante ter uma versão combinada das taxas de precisão e de sensibilidade (Equação 6.3).

$$F1 - Score = \frac{2 \times Sensibilidade \times Precisão}{Sensibilidade + Precisão} \quad (6.6)$$

- **Curva ROC:** A curva Curva Característica de Operação do Receptor (do Inglês *Receiver Operating Characteristic Curve* - ROC) (Figura 21) consiste em uma representação gráfica do desempenho de um modelo de dados quantitativos segundo onde o eixo y representa fração dos verdadeiros positivos (sensibilidade, Equação 6.3) e o eixo x representa a fração dos falsos positivos (especificidade, Equação 6.5). O valor da área sob a curva (do Inglês *area under the curve* - AUC) corresponde à área da região sob a curva ROC [4].

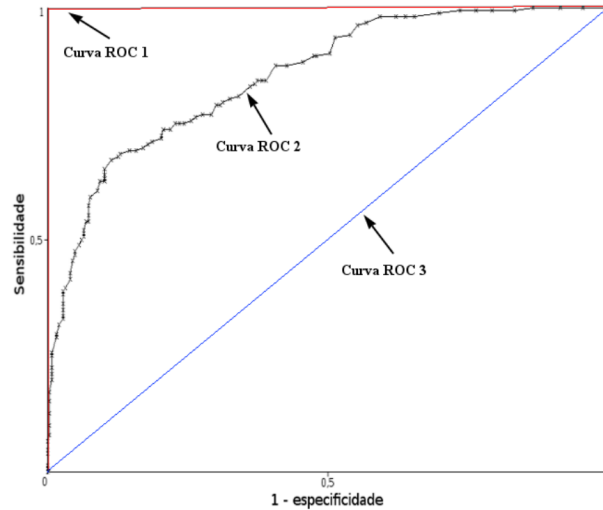


Figura 21: Exemplo de curvas ROC: a curva 1 representa um teste com desempenho perfeito de área igual a 1; a curva 2 representa um teste com desempenho intermediário de área igual a 0,75; e a curva 3 é um teste aleatório com área igual a 0,5.

Fonte: [4]

6.2.2 Resultados

Para avaliar o impacto do uso de imagens geradas artificialmente no desempenho do classificador (VGG16) vários modelos foram treinados usando diferentes quantidades de imagens sintéticas. A estratégia utilizada foi adicionar um determinado número de imagens sintéticas, divididas igualmente entre os tipos histológicos, exclusivamente, no *dataset* de treinamento da VGG16 e então avaliar os resultados utilizando um mesmo *dataset* de teste que contém apenas imagens reais. A Tabela 4 apresenta os resultados da avaliação dos modelos em relação às métricas descritas na Seção 6.2.1.

Quantidade de imagens sintéticas	Acurácia	ROC AUC	Precisão	Sensibilidade	F1-Score
0	0,906	0,909	0,899	0,909	0,903
1000	0,932	0,932	0,926	0,932	0,929
2000	0,920	0,916	0,916	0,916	0,916
3000	0,917	0,917	0,911	0,917	0,913
4000	0,940	0,940	0,936	0,940	0,938
5000	0,918	0,915	0,914	0,915	0,914
6000	0,920	0,926	0,913	0,926	0,917
7000	0,929	0,928	0,924	0,928	0,926
8000	0,902	0,901	0,895	0,901	0,898
9000	0,926	0,926	0,920	0,926	0,923
10000	0,902	0,906	0,895	0,906	0,899
11000	0,911	0,910	0,905	0,910	0,907
12000	0,894	0,892	0,888	0,892	0,890
13000	0,915	0,917	0,909	0,917	0,912
14000	0,912	0,919	0,905	0,919	0,910
15000	0,906	0,907	0,899	0,907	0,903

Tabela 4: Comparação das métricas de classificação para diferentes números de imagens sintéticas geradas pela PGGAN que foram juntamente com as imagens reais para o treinamento da VGG16. Em negrito estão os maiores valores por métrica por coluna.

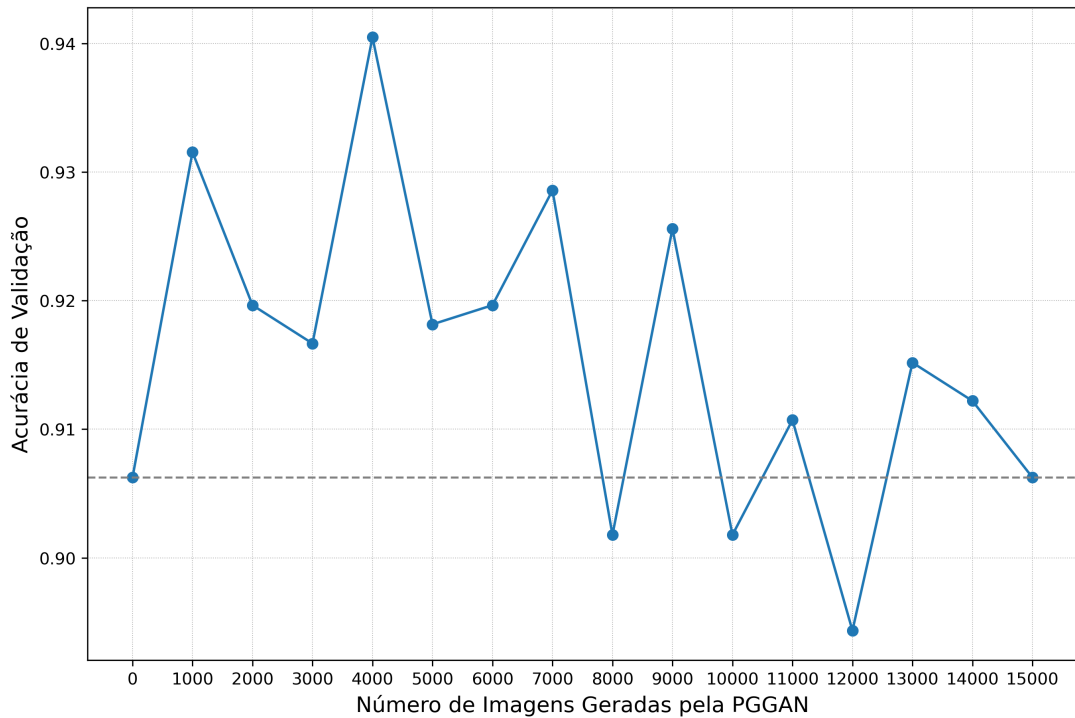


Figura 22: Avaliação da acurácia de validação para diferentes números de imagens geradas pela PGGAN que foram utilizadas no treinamento do classificador. A linha horizontal tracejada indica a acurácia do modelo treinado apenas com imagens reais. Em negrito estão os maiores valores por métrica. Fonte: Autor

Observa-se na Figura 22 que o modelo treinado com 4000 imagens geradas pela PGGAN é o que tem o melhor desempenho quando comparado com o modelo treinado apenas com imagens reais, 0.940 de acurácia contra 0.906, respectivamente, uma melhora de 3,8%. É importante notar que a partir de 6000 imagens geradas passam a existir mais imagens sintéticas do que imagens reais no *dataset* de treino, nota-se que apesar das variações na acurácia, a partir desse ponto a métrica não apresenta melhora significativa, ocorrendo inclusive casos onde a acurácia é menor que a modelo treinado apenas com imagens reais, por exemplo, para 8000, 10000 e 12000 imagens artificiais.

A queda na acurácia quando o número imagens sintéticas aumenta pode estar relacionado com o fato de que apesar das imagens geradas pela PGGAN terem um bom nível de realismo elas não são capazes de substituir completamente as imagens reais, como mostrado pelas métricas GAN-train e GAN-test na Tabela 3, pois podem haver imperfeições

e anomalias nas imagens. Dessa forma, quando o número de imagens sintéticas é relativamente pequena (próximo do número de imagens reais), as imagens geradas servem de complemento para as imagens reais, sendo capazes de cobrir pontos da distribuição de imagens reais que não são cobertos pelo dataset original, o que aumenta a variabilidade do dataset de treinamento e conseqüentemente seu desempenho, conclusão semelhante à encontrada por Han *et al.* [20]. Entretanto, quando o número de imagens sintéticas é muito grande as imperfeições e anomalias presentes nas imagens sintéticas prevalecem e o desempenho piora, tendendo a se aproximar dos resultados mostrados na Tabela 3.

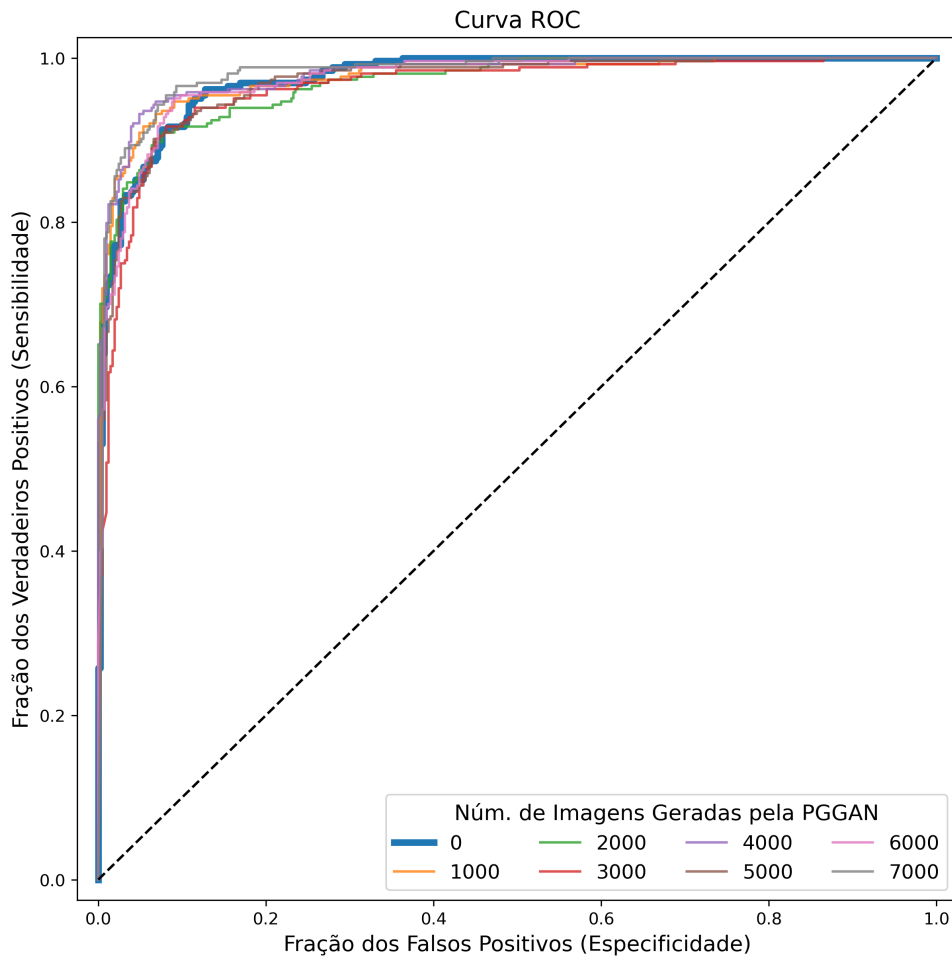


Figura 23: Comparação entre curvas ROC para diferentes números de imagens geradas pela PGGAN que foram usadas no treinamento do classificador. A curva mais espessa representa o classificador treinado apenas com imagens reais. Fonte: Autor

7 CONSIDERAÇÕES FINAIS

7.1 Trabalhos Futuros

Como trabalhos futuros, acreditamos que o uso de outros tipos de redes generativas que têm um desempenho melhor que a PGGAN como a StyleGAN2 [54], atualmente o estado da arte para a geração das imagens, pode aumentar ainda mais o desempenho do modelo. Além disso, Karras *et al.* [55] mostraram que o uso de *data augmentation* para o treinamento de GANs traz ganhos de desempenho, sendo uma outra alternativa para melhorar o desempenho do modelo. Com relação ao classificador, testar outros modelos como a ResNet [56] tem o potencial de melhorar o desempenho na classificação dos nódulos, dado que alguns estudos (como [57] e [12]) mostram que a ResNet pode ter um desempenho melhor que a VGG-16 em algumas tarefas de classificação.

REFERÊNCIAS

- 1 YUE, J. Y. et al. CT-pathologic correlation in lung adenocarcinoma and squamous cell carcinoma. *Med. (United States)*, v. 97, n. 50, p. e13362, dec 2018. ISSN 15365964.
- 2 LYRA, M.; PLOUSSI, A.; GEORGANTZOGLOU, A. MATLAB as a Tool in Nuclear Medicine Image Processing. In: *MATLAB - A Ubiquitous Tool Pract. Eng.* [S.l.]: InTech, 2011.
- 3 RAVENEL, J. *The Essential Physics of Medical Imaging*, 2nd ed. . 2. ed. Philadelphia: Lippincott Williams & Wilkins, 2003. v. 180. 596–596 p. ISSN 0361-803X.
- 4 Ferreira Junior, J. R.; SANTOS, M. K.; AZEVEDO-MARQUES, P. M. *Arcabouço para Classificação, Recuperação por Conteúdo e Radiômica de Imagens Médicas: uma investigação de biomarcadores quantitativos para o câncer de pulmão*. 13–18 p. Tese (Doutorado) — Universidade de São Paulo, São Carlos, mar 2020. Disponível em: <https://sol.sbc.org.br/index.php/sbcas{_}estendido/article/view/11>.
- 5 PATTERSON, J.; GIBSON, A. *Deep Learning. A Practitioner's Approach*. 2. ed. [S.l.: s.n.], 2017.
- 6 GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. MIT Press, 2016. Disponível em: <<http://www.deeplearningbook.org>>.
- 7 CZUM, J. M. Dive Into Deep Learning. *J. Am. Coll. Radiol.*, v. 17, n. 5, p. 637–638, jun 2020. ISSN 1558349X.
- 8 ASADI, F.; O'REILLY, J. A. Artificial Computed Tomography Images with Progressively Growing Generative Adversarial Network. *BMEiCON 2021 - 13th Biomed. Eng. Int. Conf.*, IEEE, p. 1–5, nov 2021. Disponível em: <<https://ieeexplore.ieee.org/document/9745251/>>.
- 9 HEUSEL, M. et al. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Adv. Neural Inf. Process. Syst.*, v. 2017-Decem, p. 6627–6638, jun 2017. ISSN 10495258. Disponível em: <<http://arxiv.org/abs/1706.08500>>.
- 10 SHMELKOV, K.; SCHMID, C.; ALAHARI, K. How Good Is My GAN? *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, v. 11206 LNCS, p. 218–234, jul 2018. ISSN 16113349.
- 11 Equipe Oncoguia. *Com 82% de mortalidade, câncer de pulmão é o que mais mata no Brasil*. 2018. Disponível em: <<http://www.oncoguia.org.br/conteudo/com-82-de-mortalidade-cancer-de-pulmao-e-o-que-mais-mata-no-brasil/12460/42/>>.
- 12 TOMASSINI, S. et al. Lung nodule diagnosis and cancer histology classification from computed tomography data by convolutional neural networks: A survey. *Comput. Biol. Med.*, v. 146, p. 105691, jul 2022. ISSN 18790534.

- 13 MARENTAKIS, P. et al. Lung cancer histology classification from CT images based on radiomics and deep learning models. *Med. Biol. Eng. Comput.*, v. 59, n. 1, p. 215–226, jan 2021. ISSN 17410444.
- 14 WU, C. C.; MAHER, M. M.; SHEPARD, J. A. O. Complications of CT-guided percutaneous needle biopsy of the chest: Prevention and management. *Am. J. Roentgenol.*, v. 196, n. 6, p. W678–W682, jun 2011. ISSN 0361803X.
- 15 ARIOZZI, I. et al. Computed tomography-histologic correlations in lung cancer. *Pathologica*, v. 105, n. 6, p. 329–336, dec 2013. ISSN 00312983.
- 16 COHEN, J. et al. Lung adenocarcinomas: correlation of computed tomography and pathology findings. *Diagn. Interv. Imaging*, v. 97, n. 10, p. 955–963, oct 2016. ISSN 22115684.
- 17 LITJENS, G. et al. A survey on deep learning in medical image analysis. *Med. Image Anal.*, v. 42, p. 60–88, dec 2017. ISSN 13618423.
- 18 RAHIMI, S. et al. Addressing the Exorbitant Cost of Labeling Medical Images with Active Learning. In: *Int. Conf. Mach. Learn. Med. Imaging Anal.* [s.n.], 2021. p. 9–12. Disponível em: <<https://www.microsoft.com/en-us/research/publication/addressing-the-exorbitant-cost-of-labeling-medical-images-with-active-learning/>>.
- 19 MILLETARI, F. et al. Hough-CNN: Deep learning for segmentation of deep brain regions in MRI and ultrasound. *Comput. Vis. Image Underst.*, v. 164, p. 92–102, jan 2017. ISSN 1090235X.
- 20 HAN, C. et al. Infinite Brain MR Images: PGGAN-Based Data Augmentation for Tumor Detection. *Smart Innov. Syst. Technol.*, v. 151, p. 291–303, mar 2020. ISSN 21903026. Disponível em: <<http://arxiv.org/abs/1903.12564>>.
- 21 NAIK, A.; EDLA, D. R. Lung Nodule Classification on Computed Tomography Images Using Deep Learning. *Wirel. Pers. Commun.*, v. 116, n. 1, p. 655–690, jan 2021. ISSN 1572834X.
- 22 ELSAYED, O. et al. Automatic detection of the pulmonary nodules from CT images. In: *IntelliSys 2015 - Proc. 2015 SAI Intell. Syst. Conf.* [S.l.]: IEEE, 2015. p. 742–746. ISBN 9781467376068.
- 23 TADEUSIEWICZ, R. *Neural networks: A comprehensive foundation*. 2nd. ed. USA: Prentice Hall PTR, 1995. v. 3. 746–747 p. ISSN 09670661. ISBN 0132733501.
- 24 MATHWORKS. *How Do Neural Networks Work?* 2019. 1 p. Disponível em: <<https://www.mathworks.com/discovery/neural-network.html>>.
- 25 SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, sep 2015.
- 26 BORJI, A. Pros and cons of GAN evaluation measures. *Comput. Vis. Image Underst.*, v. 179, p. 41–65, feb 2019. ISSN 1090235X.

- 27 GOODFELLOW, I. et al. Generative adversarial networks. *Commun. ACM*, v. 63, n. 11, p. 139–144, oct 2020. ISSN 15577317. Disponível em: <<https://dl.acm.org/doi/10.1145/3422622>>.
- 28 SKANDARANI, Y.; JODOIN, P.-M.; LALANDE, A. GANs for Medical Image Synthesis: An Empirical Study. may 2021. Disponível em: <<http://arxiv.org/abs/2105.05318>>.
- 29 KARRAS, T. et al. Progressive growing of GANs for improved quality, stability, and variation. *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, oct 2018.
- 30 ARJOVSKY, M.; CHINTALA, S.; BOTTOU, L. Wasserstein GAN. jan 2017. Disponível em: <<http://arxiv.org/abs/1701.07875>>.
- 31 GULRAJANI, I. et al. Improved training of wasserstein GANs. *Adv. Neural Inf. Process. Syst.*, v. 2017-Decem, p. 5768–5778, mar 2017. ISSN 10495258. Disponível em: <<http://arxiv.org/abs/1704.00028>>.
- 32 PEREIRA, T. *Por Que a Linguagem Python é Tão Popular em Machine Learning e Inteligência Artificial?* 2020. Disponível em: <<http://datascienceacademy.com.br/blog/por-que-a-linguagem-python-e-tao-popular-em-machine-learning-e-inteligencia-artificial/>>.
- 33 Müller, A., & Guido, S. *Introduction to Machine Learning with Python: A Guide for Data Scientists 1st Edition*. O'Reilly Media. [S.l.]: O'Reilly Media, 2018. v. 53. 378 p. ISSN 1098-6596. ISBN 1449369413.
- 34 MCKINNEY, W. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. 2nd. ed. O'Reilly Media, Inc., 2012. 466 p. ISSN 02705257. ISBN 1449323618. Disponível em: <[https://books.google.com/books?id=v3n4{_}AK8vu0C{&}](https://books.google.com/books?id=v3n4{_}AK8vu0C{&}>)>.
- 35 HUNTER, J. et al. *Matplotlib: Visualization with Python*. 2020. Disponível em: <<https://matplotlib.org/>>.
- 36 O'CONNOR, R. *PyTorch vs TensorFlow in 2022*. 2021. NA p. Disponível em: <<https://www.assemblyai.com/blog/pytorch-vs-tensorflow-in-2022/>>.
- 37 PYTORCH. *Tensors — PyTorch Tutorials 1.13.0+cu117 documentation*. 2022. Disponível em: <https://pytorch.org/tutorials/beginner/blitz/tensor{_}tutorial.h>.
- 38 NVIDIA. *NVIDIA T4 Tensor Core GPU for AI Inference | NVIDIA Data Center*. Disponível em: <<https://www.nvidia.com/en-us/data-center/tesla-t4/>>.
- 39 MUSTRA, M.; DELAC, K.; GRGIC, M. Overview of the DICOM standard. In: *Proc. Elmar - Int. Symp. Electron. Mar.* [S.l.: s.n.], 2008. v. 1, p. 39–44. ISSN 13342630.
- 40 FEDOROV, A. et al. 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magn. Reson. Imaging*, v. 30, n. 9, p. 1323–1341, nov 2012. ISSN 0730725X. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0730725X12001816>>.
- 41 LI, P. et al. A Large-Scale CT and PET/CT Dataset for Lung Cancer Diagnosis. *Cancer Imaging Arch.*, 2020. Disponível em: <<https://doi.org/10.7937/TCIA.2020.NNC2-0461>>.

- 42 CLARK, K. et al. The cancer imaging archive (TCIA): Maintaining and operating a public information repository. *J. Digit. Imaging*, v. 26, n. 6, p. 1045–1057, dec 2013. ISSN 08971889. Disponível em: <<http://link.springer.com/10.1007/s10278-013-9622-7>>.
- 43 AERTS, H. J. et al. Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach. *Nat. Commun.*, v. 5, n. 1, p. 4006, sep 2014. ISSN 20411723. Disponível em: <<http://www.nature.com/articles/ncomms5006>>.
- 44 AERTS, H. J. W. L. et al. Dados do NSCLC-Radiomics [Data set]. *Cancer Imaging Arch.*, 2019. Disponível em: <<https://doi.org/10.7937/K9/TCIA.2015.PF0M9REI>>.
- 45 PANG, S. et al. VGG16-T: A novel deep convolutional neural network with boosting to identify pathological type of lung cancer in early stage by ct images. *Int. J. Comput. Intell. Syst.*, v. 13, n. 1, p. 771–780, 2020. ISSN 18756883.
- 46 ZHAO, D. et al. Synthetic medical images using F & BGANfor improved lung nodules classification bymulti-scale VGG16. *Symmetry (Basel)*., v. 10, n. 10, p. 519, oct 2018. ISSN 20738994.
- 47 XIAO, Z. et al. Segmentation of lung nodules using improved 3D-UNet neural network. *Symmetry (Basel)*., v. 12, n. 11, p. 1–15, oct 2020. ISSN 20738994.
- 48 KINGMA, D. P.; BA, J. Adam: A Method for Stochastic Optimization. dec 2014. Disponível em: <<http://arxiv.org/abs/1412.6980>>.
- 49 FERNANDES, A. A. T. et al. Read this paper if you want to learn logistic regression. *Rev. Sociol. e Política*, v. 28, n. 74, 2020. ISSN 1678-9873.
- 50 SRIVASTAVA, N. et al. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, v. 15, p. 1929–1958, 2014. ISSN 15337928.
- 51 SALIMANS, T. et al. Improved techniques for training GANs. In: LEE, D. et al. (Ed.). *Adv. Neural Inf. Process. Syst.* Curran Associates, Inc., 2016. v. 29, p. 2234–2242. ISSN 10495258. Disponível em: <<https://proceedings.neurips.cc/paper/2016/file/8a3363abe792db2d8761d6403605aeb7-Paper.pdf>>.
- 52 DIMITRAKOPOULOS, P.; SFIKAS, G.; NIKOU, C. Wind: Wasserstein Inception Distance for Evaluating Generative Adversarial Network Performance. In: *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc. IEEE*, 2020. v. 2020-May, p. 3182–3186. ISBN 9781509066315. ISSN 15206149. Disponível em: <<https://ieeexplore.ieee.org/document/9053325/>>.
- 53 MINAEI, S. et al. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 44, n. 7, p. 3523–3542, 2022. ISSN 19393539.
- 54 KARRAS, T. et al. Analyzing and Improving the Image Quality of StyleGAN. dec 2019. Disponível em: <<http://arxiv.org/abs/1912.04958>>.
- 55 KARRAS, T. et al. Training Generative Adversarial Networks with Limited Data. jun 2020. Disponível em: <<http://arxiv.org/abs/2006.06676>>.
- 56 HE, K. et al. Deep Residual Learning for Image Recognition. dec 2015. Disponível em: <<http://arxiv.org/abs/1512.03385>>.

57 ZHAO, P. et al. A Comparative Study of Deep Learning Classification Methods on a Small Environmental Microorganism Image Dataset (EMDS-6): From Convolutional Neural Networks to Visual Transformers. *Front. Microbiol.*, v. 13, mar 2022. ISSN 1664-302X. Disponível em: <<https://www.frontiersin.org/articles/10.3389/fmicb.2022.792166/full>>.