Classificação de Lesões Histopatológicas Colorretais Utilizando Redes Neurais Convolucionais

Antonio Borssato
Departamento de Informática
Universidade Federal do Espírito Santo
Vitória, Brasil
antonio.borssato@edu.ufes.br

Lucas Alves
Departamento de Informática
Universidade Federal do Espírito Santo
Vitória, Brasil
lucas.o.alves@edu.ufes.br

Rodrigo Fardin
Departamento de Informática
Universidade Federal do Espírito Santo
Vitória, Brasil
rodrigo.fardin@edu.ufes.br

***Resumo* — Este estudo apresenta o desenvolvimento e avaliação de um pipeline focado na classificação de lesões em imagens histopatológicas utilizando Redes Neurais Convolucionais (CNNs). O conjunto de dados empregado foi o MHIST (Minimalist Histopathology Image Analysis Dataset), desenvolvido pelo Departamento de Patologia e Medicina Laboratorial do Centro Médico Dartmouth-Hitchcock. O objetivo principal foi explorar e comparar o desempenho de diferentes arquiteturas de CNN, incluindo uma arquitetura customizada simples (SimpleCNN) desenvolvida pelos autores e modelos pré-treinados de ponta como ResNet50, VGG16 e DenseNet121, por meio de transfer learning e fine-tuning, no desafio de classificar corretamente as imagens entre os dois tipos de pólipos colorretais, sendo um benigno e outro pré-canceroso. O melhor modelo individual foi a rede pré-treinada ResNet50 com acurácia de 85% e AUC de 91.31%, entretanto, o ensemble softmax ponderado dos modelos superou esses valor com acurácia de 86% AUC de 93.12%. Os resultados demonstram a viabilidade do uso de CNNs para a classificação de lesões em imagens histopatológicas.**

***Palavras-chave* — Histopatologia, Classificação de imagens, Redes Neurais Convolucionais (CNNs), Transfer learning, Fine-tuning, MHIST**

I. Introdução

O microscópio, do grego micro (pequeno) e skopein (olhar) [13], foi feito pela primeira vez no século XVI por Zacharias Janssen e seu pai, Hans Janssen, com poder de aumento de 9x. Posteriormente, foi aprimorado por pesquisadores como Robert Hooke (1635-1703) e Antonie Van Leeuwenhoek (1632-1723), que descobriram novas possibilidades de estudo [16].

Na medicina, o microscópio tornou-se essencial para compreender as bases celulares das doenças [16]. Entretanto, a interpretação de imagens histológicas exige grande destreza nas técnicas de microscopia para identificar as estruturas e analisar os tecidos adjacentes [12]. Além do mais, a mudança para o ensino médico baseado em resolução de problemas (PBL) consolidou a necessidade de tecnologias para otimização do tempo em laboratório, o que abriu espaço para a microscopia digital, modelagem computacional e o uso das redes neurais artificiais, principalmente do tipo convolucional (CNNs) [14].

As CNNs se destacaram nesses cenários, produzindo resultados comparáveis aos humanos [15]. No entanto, o ramo da análise de imagens histopatológicas enfrenta desafios como a escassez de dados para treinar modelos, dificuldade de anotação e o desafio para obter aprovação para divulgação [1]. Para superar essas barreiras, o dataset MHIST foi desenvolvido como um conjunto de dados minimalista e de fácil uso para a classificação binária de pólipos colorretais [1].

Em relação aos pólipos colorretais, que são crescimentos na parede interna do cólon ou do reto, a maioria dessas projeções é considerada benigna, porém, alguns pólipos podem caracterizar lesões pré-cancerosas [2]. Com isso, a detecção precoce de lesões malignas é fundamental para prevenir o câncer colorretal, que é um dos tipos de câncer com maior mortalidade no mundo [2], [5]. Nesse contexto, a diferenciação entre pólipos hiperplásicos (HP), benignos, e adenomas serrilhados sésseis (SSA), pré-cancerosos, é uma das tarefas de maior relevância e volume na prática patológica [1].

Diante dessa necessidade, o uso de CNNs aplicadas ao dataset MHIST surge como uma abordagem promissora para automatizar e auxiliar a prática clínica. Assim, este trabalho propõe e avalia um pipeline de classificação de imagens histopatológicas através de CNNs, usando o dataset MHIST como ”placa de petri”. O objetivo é comparar o desempenho de diferentes arquiteturas e investigar o impacto de técnicas de ensemble de modelos na qualidade da classificação.

II. Revisão da Literatura

Por ser uma base de dados minimalista, o MHIST já foi usado em outros trabalhos de comparação para análise de imagens histopatológicas. A primeira pesquisa realizada com o MHIST foi a de Wei et al. [1]. Nele, os autores introduzem o conjunto de dados e mostram treinamento de algumas CNNs, como por exemplo a ResNet50, alcançando AUC igual a 83.0% ± 0.6. Kang et al. [17] também realizaram um estudo através do MHIST, utilizando o Self-supervised learning (SSL). Seu objetivo era avaliar quatro métodos representativos de SSL usando dados pré-treinados da área médica, onde o MHIST foi aplicado como dataset downstream. O melhor resultado encontrado foi também com a ResNet50, que alcançou acurácia de 85.88%.

No estudo ”An automatic classification method of testicular histopathology based on SC-YOLO framework”, Wu et al. [18] apresentam uma versão aprimorada do modelo YOLO (You Only Look Once), incorporando módulos S3Ghost, CoordAtt e DCNv2. Nesse caso, o MHIST é usado como o conjunto de

imagens histopatológicas bem anotadas, compartilhando com o YOLO o desafio de classificação de imagens patológicas. Os resultados foram de 84% de acurácia, 83% de precisão e 84% de recall.

Com base na literatura revisada, o desempenho de modelos de CNN no dataset MHIST geralmente varia entre 80% e 90% para acurácia e AUC. Esses valores servem de base para comparação com os nossos resultados. Além disso, a arquitetura pré-treinada ResNet50 se destaca, demonstrando bom desempenho em diversos artigos. Por isso, essa rede será avaliada em nosso trabalho.

III. Conjunto de Dados

A base de dados utilizada neste trabalho é a MHIST (Minimalist Histopathology Image Analysis Dataset) [1], desenvolvida no Dartmouth-Hitchcock Medical Center (DHMC) para servir como benchmark de análise de imagens histopatológicas. O dataset é composto por 3152 imagens de pólipos colorretais de tamanho fixo (224x224 pixels). A tarefa consiste na classificação binária de pólipos benignos (Hiperplásicos - HP) e pré-cancerosos (Adenomas Serrilhados Sésseis - SSA) [2]. A classe de cada imagem foi definida por votação majoritária de sete patologistas gastrointestinais [1]. A distribuição de registros por classe é detalhada na Tabela I:

Tabela I
Proporção das classes na base de dados

Classe	nº de imagens	%
HP	2162	68.59
SSA	990	31.41
Total	**3152**	**100**

Todas as imagens são de extrações de tecido fixado em formalina e embebido em parafina, coradas com hematoxilina e eosina. O dataset não contém dados pessoais de pacientes e sua divulgação foi aprovada pelo Comitê de Ética em Pesquisa da Dartmouth-Hitchcock Health [1]. A Tabela II apresenta a distribuição de imagens para as partições de treino e validação, conforme fornecido pela própria base de dados.

Tabela II
Proporção das imagens de treino e teste

Partição	nº de imagens	%
Treino	2175	69,0
Validação	977	31,0
Total	**3152**	**100**

IV. Metodologia

A abordagem metodológica deste estudo foi estruturada utilizando o PyTorch com as seguintes etapas:

A. Carregamento e Pré-processamento dos dados

Inicialmente, os dados foram carregados juntamente com as anotações baseadas no voto majoritário dos sete patologistas. Para aumentar a variabilidade dos dados e a capacidade de generalização dos modelos, foram definidas estratégias de transformação e data augmentation. As seguintes transformações foram aplicadas:

- **Dataset de Treino**: Todas as imagens passaram por um recorte aleatório que manteve de 80% a 100% da área original. O resultado dessa operação foi reescalado para 224x224 pixels, mantendo as proporções originais das imagens. Para aumentar a variabilidade e a robustez, foram aplicadas as seguintes transformações: flip horizontal (50% das amostras), rotação aleatória de até 10 graus, e perturbações de cor controladas (brilho, contraste, saturação e matiz). Distorções geométricas de baixa intensidade foram inseridas com 20% de probabilidade. Por fim, os canais RGB foram normalizados usando as médias e desvios padrão da ImageNet(mean=[0.485; 0.456; 0.406], std=[0.229; 0.224; 0.225]).
- **Dataset de Teste**: Apenas a normalização dos canais RGB usando as médias e desvios padrão da imagenet.

A Fig. 1 mostra dois exemplos das imagens originais e transformadas do dataset de treino.

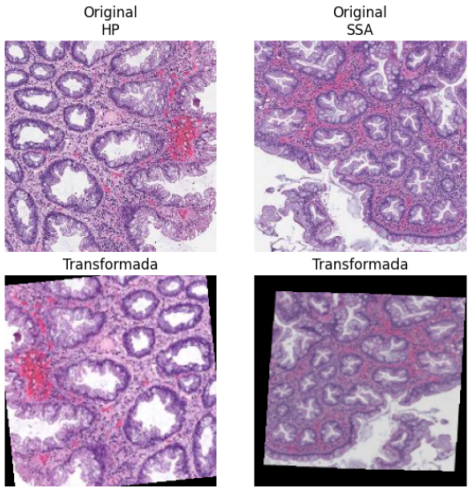

Figura 1. Exemplos de imagens originais e transformadas do dataset de treino)

Os dados foram carregados em batches de 32, garantindo a leitura no espaço de cor RGB e a aplicação das transformações definidas. Para lidar com o desbalanceamento entre as classes HP e SSA, foi utilizada uma estratégia de amostragem ponderada no treinamento. Os pesos de amostragem foram calculados de forma inversamente proporcional à frequência de cada classe, garantindo que o rótulo minoritário tivesse maior probabilidade de ser selecionado, equilibrando o treinamento. A equação (1) expressa o cálculo usado:

$$w_i = \frac{N}{2 \cdot n_i} \tag{1}$$

onde:
- w_i é o peso da classe i;
- N é o número total de amostras no dataset;
- n_i é o número de amostras da classe i.

B. Definição dos Modelos

1) SimpleCNN: A arquitetura da CNN seguiu uma estrutura simples de blocos de convolução para classificação de imagens. A rede recebe imagens RGB de 224x224 pixels, que passam por uma sequência de quatro blocos, cada um composto por uma camada de convolução 2D com kernels 3x3. Os blocos 1, 2, 3 e 4 têm camadas de convolução com respectivamente 32, 64, 128 e 256 filtros. Em seguida há o batch normalization, ativação ReLU e max pooling com janela 2×2 e stride 2. Isso resulta em um tensor de saída de dimensão 256x14x14, que é achatado em um vetor de 50176 elementos. O resultado é passado para o classificador, que é uma rede totalmente conectada, composta por três camadas lineares que mapeiam de 50176 para 512, 512 para 256 e 256 para 2, que a quantidade de classe do problema. As duas primeiras camadas lineares incluem batch normalization, ReLU e dropout (taxa de 0.4) para acelerar a convergência e reduzir o sobreajuste. A camada de saída produz os logits para a classificação final.

Na Tabela III temos a arquitetura completa da SimpleCNN, com cada camada dos blocos convolucionais (Conv) e o bloco totalmente conectado (FC):

Tabela III
ARQUITETURA DA SIMPLECNN

Bloco	Camada	Tipo	Output Shape	Param #
Conv 1	1	Conv2d	[32, 32, 224, 224]	896
	2	BatchNorm2d	[32, 32, 224, 224]	64
	3	ReLU	[32, 32, 224, 224]	0
	4	MaxPool2d	[32, 32, 112, 112]	0
Conv 2	5	Conv2d	[32, 64, 112, 112]	18,496
	6	BatchNorm2d	[32, 64, 112, 112]	128
	7	ReLU	[32, 64, 112, 112]	0
	8	MaxPool2d	[32, 64, 56, 56]	0
Conv 3	9	Conv2d	[32, 128, 56, 56]	73,856
	10	BatchNorm2d	[32, 128, 56, 56]	256
	11	ReLU	[32, 128, 56, 56]	0
	12	MaxPool2d	[32, 128, 28, 28]	0
Conv 4	13	Conv2d	[32, 256, 28, 28]	295,168
	14	BatchNorm2d	[32, 256, 28, 28]	512
	15	ReLU	[32, 256, 28, 28]	0
	16	MaxPool2d	[32, 256, 14, 14]	0
FC	17	Linear	[32, 512]	25,690,624
	18	BatchNorm1d	[32, 512]	1,024
	19	ReLU	[32, 512]	0
	20	Dropout	[32, 512]	0
	21	Linear	[32, 256]	131,328
	22	BatchNorm1d	[32, 256]	512
	23	ReLU	[32, 256]	0
	24	Dropout	[32, 256]	0
	25	Linear (out)	[32, 2]	514

2) ResNet50: A ResNet50 funciona como o backbone convolucional pré-treinado (pesos padrão pré-treinados em ImageNet foram carregados). A ResNet-50 é composta por uma sequência de blocos residuais do tipo bottleneck com conexões de atalho que permitem o treinamento estável de redes profundas. Esses blocos extraem representações hierárquicas, terminando em um global average pooling que produz um vetor de característica de dimensão 2048 por amostra [6].

No nosso modelo, parte do backbone foi mantida congelada (sem atualização de pesos) e a cabeça de classificação original foi substituída por uma camada linear que mapeia o vetor de 2048 dimensões para duas classes. Em resumo, das quatro camadas profundas (layers 1–4) e da camada totalmente conectada (FC) da ResNet, descongelamos as layers 3 e 4 e a FC. Essa estratégia de transfer learning aproveita o conhecimento pré-treinado da ResNet50 e permite a adaptação ao domínio específico dos dados do MHIST.

Para consistência no input do treinamento, as imagens foram inseridas com os mesmos padrões de pré-processamento da SimpleCNN, também em batches de tamanho 32.

A Fig. 2 apresenta uma representação didática da arquitetura ResNet50.

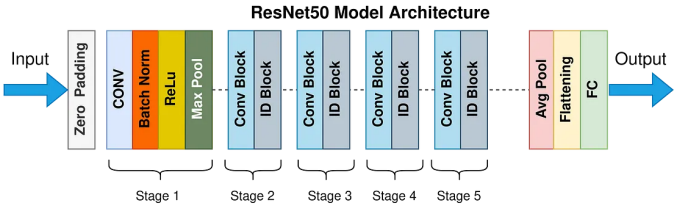

Figura 2. Representação simplificada da arquitetura ResNet50 [9]

3) VGG16: Assim como na ResNest, a VGG-16 foi utilizada como backbone convolucional pré-treinado (pesos padrão também obtidos por pré-treinamento no ImageNet foram carregados). A arquitetura VGG-16 caracteriza-se por pilhas de cinco blocos convolucionais com kernels 3×3 de pequena receptividade intercaladas por operações de max-pooling, seguida por um classificador totalmente conectado composto originalmente por três camadas densas com ativação por ReLU e dropout para regularização [7].

No nosso modelo, a maior parte do backbone foi mantida congelada durante o treino, enquanto o último bloco convolucional (bloco conv5) e a última camada do classificador totalmente conectado (layer 3) foram treináveis. A cabeça de classificação original foi adaptada substituindo a última camada totalmente conectada por uma nova camada linear com saída para duas classes. Com isso, reduz-se a quantidade de parâmetros treináveis enquanto se beneficia do fine-tuning das camadas finais.

Para consistência, as imagens de entrada seguiram o mesmo pré-processamento aplicado aos outros modelos e foram processadas em batches de tamanho 32.

A Fig. 3 apresenta uma representação didática da arquitetura VGG16.

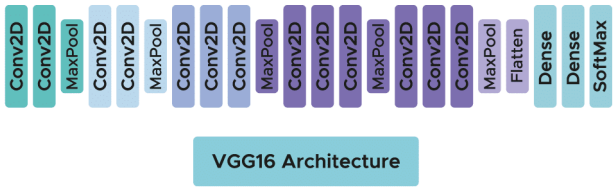

Figura 3. Representação simplificada da arquitetura VGG16 [10]

4) DenseNet121: Assim como nos modelos anteriores, a DenseNet121 funciona como o backbone convolucional pré-

treinado, carregado com pesos do ImageNet. Em relação à arquitetura da rede, a DenseNet121 é composta por uma sequência de quatro blocos densos, nos quais cada camada recebe como entrada todas as ativações das camadas anteriores dentro do bloco, promovendo reuso de características e melhor fluxo de gradiente. Entre os blocos densos há camadas de transição que realizam convoluções e pooling, reduzindo a dimensão espacial. Ao final, é feita uma normalização e um global average pooling que produz um vetor de características de dimensão 1024 por amostra, que é então fornecido para o classificador linear final, originalmente com 1000 classes [8].

No nosso modelo, a maior parte do backbone foi mantida congelada durante o treinamento, enquanto o último bloco denso (denseblock4), a camada de normalização final (norm5) e o classificador foram treináveis. A camada de classificação original foi substituída por uma nova camada linear que mapeia o vetor de 1024 dimensões para as duas classes da tarefa, em vez de 1000. Essa estratégia, assim como nos modelos anteriores, serve para reduzir o número de parâmetros treináveis e usufruir de transfer learning, ao mesmo tempo que permite fine-tuning específico aos dados do MHIST.

Para consistência no input do treinamento, as imagens também foram inseridas com os mesmos padrões de pré-processamento anteriomente citados, em batches de tamanho 32.

A Fig. 4 apresenta uma representação didática da arquitetura DenseNet121.

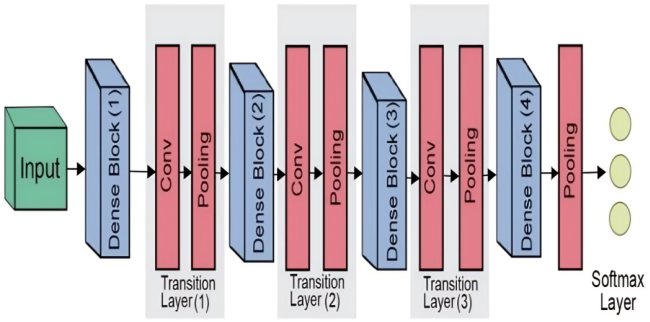

Figura 4. Representação simplificada da arquitetura DenseNet121 [11]

Na Tabela IV é possível observar a complexidade de treinamento de cada modelo. As porcentagens de parâmetros treináveis indicadas correspondem às configurações que, nos testes, obtiveram o melhor desempenho.

Tabela IV
COMPLEXIDADE DE TREINAMENTO DOS MODELOS

Modelo	Parâm. Treináveis	Total de Parâm.	% Treinável
SimpleCNN	26,213,378	26,213,378	100.00%
ResNet50	22,067,202	23,512,130	93.85%
VGG16	7,087,618	134,268,738	5.28%
DenseNet121	2,162,178	6,955,906	31.08%

C. Treinamento dos Modelos

O treinamento dos modelos foi realizado com batches de 32 imagens pré-processadas, considerando o desbalanceamento das classes HP e SSA, conforme apresentado em IV-A, garantindo que o modelo recebesse aproximadamente a mesma quantidade de exemplos de cada classe durante cada época.

A função de perda utilizada foi a Cross Entropy Loss, ponderada pelos pesos das classes, de forma a penalizar mais fortemente erros em classes menos representadas. O otimizador adotado foi o AdamW com weight decay igual a 3×10^{-4}, que é uma técnica de regularização L2 usada para evitar overfitting. O otimizador foi aplicado apenas aos parâmetros treináveis de cada modelo, conforme definido para cada arquitetura em IV-B.

O aprendizado, por sua vez, foi controlado por um scheduler do tipo 1cycle learning rate, que ajusta dinamicamente a taxa de aprendizado (learning rate) ao longo das iterações do treinamento [3], [4]. O scheduler foi configurado com um warm-up inicial em 30% das iterações, durante o qual o learning rate aumenta gradualmente a partir de um valor inicial equivalente a 1/10 do valor base definido (base_lr). Após o pico, o learning rate decai suavemente até um valor final igual a 1/1000 do valor inicial, permitindo ajustes finos dos pesos e estabilidade no treinamento. Além do mais, os modelos pré-treinados receberam um learning rate equivalente a 10% do valor fornecido ao SimpleCNN, para que fossem mais estáveis durante o treino.

O processo de treinamento de cada época consistiu em um forward pass seguido do cálculo da perda, backpropagation e atualização dos pesos treináveis. Durante a execução, foram monitoradas métricas de loss e acurácia, tanto para o conjunto de treinamento quanto para o conjunto de validação. A validação foi realizada sem cálculo de gradientes, registrando-se acurácia e perda, bem como os valores previstos pelos modelos, permitindo análise posterior do desempenho.

A etapa de treinamento foi realizada por 50 épocas, ou até a convergência, para todos os modelos, mantendo consistência nas métricas e comparação direta entre diferentes arquiteturas. Como forma de evitar overfitting, foi adotado um early stopping com paciência de 10 épocas, considerando a métrica de acurácia da validação. Isso significa que caso o modelo não melhorasse a métrica em 10 épocas, o treinamento era encerrado, salvando o melhor modelo. Na Tabela V podemos ver um resumo dos parâmetros usados por cada modelo no treinamento:

Tabela V
RESUMO DOS PARÂMETROS DE TREINAMENTO DOS MODELOS

Modelo	batch size	Épocas	Base_lr	Weight Decay
SimpleCNN	32	50	1×10^{-3}	3×10^{-4}
ResNet50	32	50	1×10^{-4}	3×10^{-4}
VGG16	32	50	1×10^{-4}	3×10^{-4}
DenseNet121	32	50	1×10^{-4}	3×10^{-4}

D. Métricas de Avalição

As métricas utilizadas para avaliar o desempenho dos modelos foram Acurácia, adotada como critério para o early stopping durante o treinamento, além de Precisão, Revocação, F1-Score e AUC. Essas métricas permitem uma análise mais

completa do desempenho, especialmente em um cenário de classes desbalanceadas, e foram aplicadas na comparação entre os quatro modelos testados no dataset MHIST.

V. Resultados

Apresentamos nessa seção os resultados quantitativos e observações qualitativas dos quatro modelos avaliados e de dois esquemas de ensemble (softmax ponderado com 4 modelos e votação Top-3 modelos).

A. *Curvas de aprendizado*

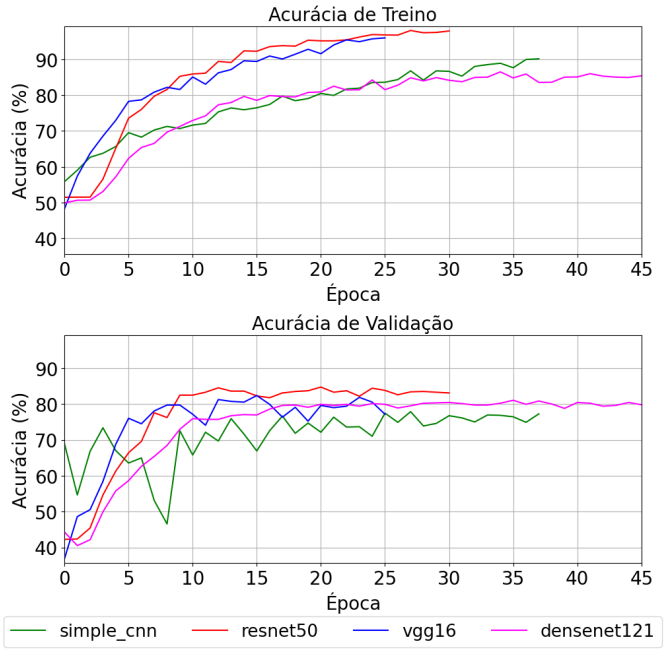

Figura 5. Curva de acurácia no treino e validação para todos os modelos.

Nota-se nas curvas de aprendizado da Fig. 5 (Acurácia de Treino e Validação) um possível comportamento de overfitting nos modelos VGG16 e ResNet50, que foram os modelos que convergiram mais rápido. Embora a acurácia de treinamento para ambos tenha se aproximado de 100%, a acurácia de validação estagnou em valores mais baixos, próximos de 80%. Um comportamento semelhante também é observado no modelo SimpleCNN, onde a acurácia cai consideravelmente do treino para validação. Quanto à estabilidade, todos os treinamentos, exceto o VGG16 e SimpleCNN, apresentaram comportamento estável, principalmente a DenseNet121.

Em relação às curvas de perda da Fig. 6, os resultados reforçam as observações de overfitting. A perda de validação para os modelos ResNet50 e VGG16 estagna ou começa a subir após aproximadamente 10 épocas, enquanto a perda de treino continua a diminuir. O SimpleCNN mostra uma maior volatilidade na perda de validação, sugerindo menor estabilidade no processo de otimização. Em contrapartida, a perda de validação do DenseNet121 decai de forma mais consistente, mostrando estabilidade no treinamento e bom ajuste.

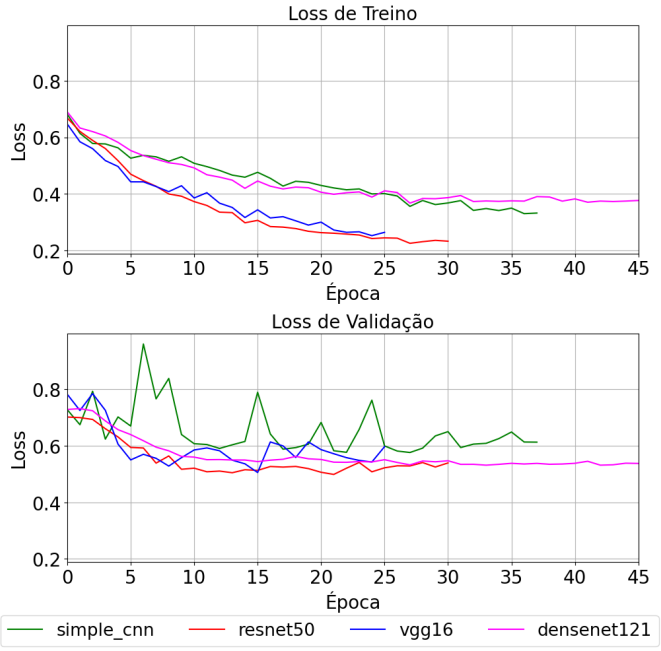

Figura 6. Curva de loss no treino e validação para todos os modelos.

Na Fig. 7 podemos ver as matrizes de confusão de cada modelo no dataset de validação.

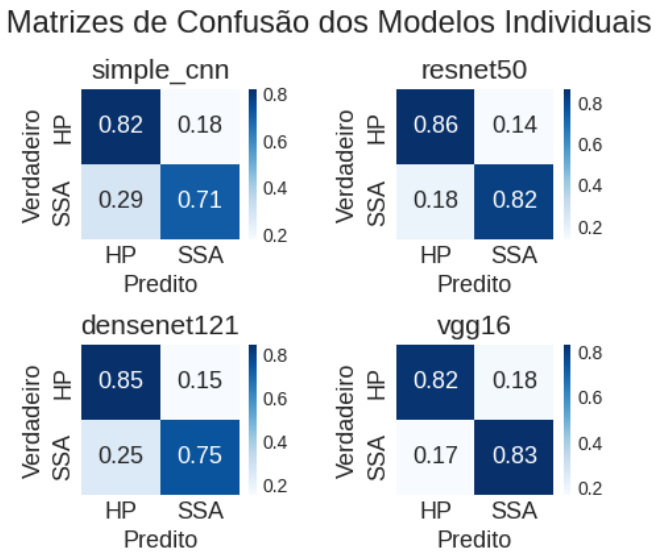

Figura 7. Matrizes de confusão de cada modelo, normalizadas por linha.

Nota-se que os modelos que sugerem overfitting, ResNet50 e VGG16, apresentam os melhores valores de verdadeiros positivos e verdadeiros negativos, sendo que o ResNet50, que previu corretamente 86% dos casos HP e VGG16 previu corretamente 82% dos casos SSA.

A taxa de falsos negativos (Casos SSA incorretamente classificados como HP e vice-versa) é um ponto crítico, pois a classificação incorreta de um pólipo SSA como HP pode levar

a um diagnóstico e tratamento inadequados que pode evoluir para câncer nos pacientes. Nesse aspecto, o SimpleCNN apresentou a maior taxa de erro, com 29% de SSA predito como HP. Em contraste, ResNet50 e VGG16 demonstraram as menores taxas de com 18% e 17%, respectivamente. Percebe-se, então, que os modelos com possível overfitting ainda assim apresentam o desempenho mais elevado no dataset de validação.

Como forma de avaliação extra e tentativa de observar a combinação de modelos com possível overfitting e modelos que apresentaram treinamento estável, fizemos dois ensembles, sendo um ponderado pelo desempenho dos quatro modelos no processo de treinamento e outro considerando a predição dos três melhores modelos em termos de acurácia. O ensemble ponderado calcula a média ponderada das probabilidades de saída (softmax) de cada modelo. A ponderação é baseada no quadrado da acurácia de validação de cada modelo, normalizada para que a soma dos pesos seja 1. Já o ensemble top 3 utiliza um método de votação majoritária, onde a classe da imagem será aquela com maior número de voto entre os três modelos, ResNet50, VGG16 e DenseNet121. Na Tabela VI encontram-se os pesos de cada modelo para o ensemble ponderado.

Tabela VI
PESOS NORMALIZADOS PARA O ENSEMBLE SOFTMAX PONDERADO

Modelo	Acurácia (%)	Peso Normalizado
SimpleCNN	77.89	0.2280
ResNet50	84.75	0.2699
DenseNet121	81.06	0.2470
VGG16	82.40	0.2551

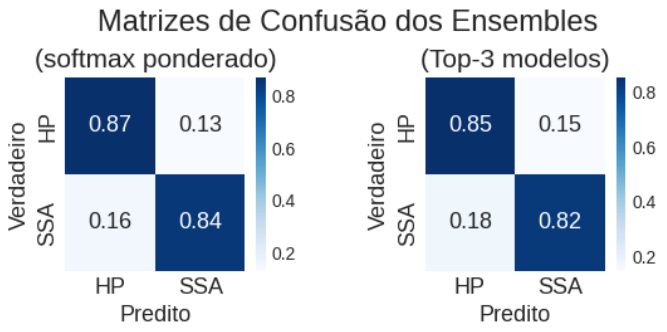

Figura 8. Matrizes dos ensembles ponderado e top3 melhores modelos, normalizadas por linha.

Percebe-se da matriz de confusão da Fig. 8 uma ótima eficácia da estratégia de ensemble, principalmente na abordagem ponderada, que supera todos os modelos individuais.

Na Tabela VII temos o compilado do desempenho de todos os modelos e ensembles e na Tabela VIII encontram-se os valores de AUC.

VI. DISCUSSÃO

A avaliação individual mostrou que as arquiteturas pré-treinadas superaram a SimpleCNN, com destaque para a ResNet50, que atingiu 85% de acurácia e 91.31% de AUC. Apesar disso, as curvas de aprendizado (Fig.5 e Fig.6) indicam overfitting tanto na ResNet50 quanto na VGG16: a acurácia de treino chegou a quase 100%, enquanto a de validação estagnou em valores menores. Ainda assim, ambos obtiveram bons resultados no conjunto de validação. A Tabela VIII e as matrizes de confusão (Fig. 7) confirmam que ResNet50 e VGG16 minimizaram melhor os erros mais críticos (falsos negativos) em comparação à SimpleCNN, que apresentou a maior taxa de erro.

Entretanto, vale salientar que, embora modelos em overfitting possam manter bom desempenho na validação, eles correm risco de falhar em dados fora dessa distribuição. Nesse caso, o pré-treinamento em grandes bases como o ImageNet ajudou a extrair características relevantes e pode explicar os resultados superiores.

Para mitigar limitações individuais, adotou-se a estratégia de ensemble. O Ensemble Softmax Ponderado foi o mais eficaz, alcançando 86% de acurácia e 93.12% de AUC, superando os modelos isolados.

Em síntese, os resultados desse trabalho se aproximam daqueles observados na literatura. Arquiteturas pré-treinadas mostraram-se eficazes, ainda que com sinais de overfitting, e a abordagem de ensemble se destacou como solução mais robusta ao combinar forças individuais e reduzir fraquezas.

Tabela VII
DESEMPENHO DOS MODELOS DE CLASSIFICAÇÃO POR CLASSE

Modelo	Classe	Acurácia	Precisão	Recall	F1-Score
SimpleCNN	HP	0.78	0.83	0.82	0.82
	SSA	0.78	0.70	0.71	0.70
ResNet50	HP	0.85	0.89	0.86	0.88
	SSA	0.85	0.78	0.82	0.80
VGG16	HP	0.82	0.89	0.82	0.85
	SSA	0.82	0.73	0.83	0.78
DenseNet121	HP	0.81	0.85	0.85	0.85
	SSA	0.81	0.74	0.75	0.74
Ensemble Softmax Ponderado	HP	**0.86**	**0.90**	**0.87**	**0.89**
	SSA	**0.86**	**0.79**	**0.84**	**0.81**
Ensemble Top-3 Melhores	HP	0.84	0.89	0.85	0.87
	SSA	0.84	0.76	0.82	0.79

Tabela VIII
AUC DOS MODELOS E ENSEMBLES

Modelo	AUC (%)
SimpleCNN	85.77
ResNet50	91.31
VGG16	90.42
DenseNet121	88.38
Ensemble Softmax Ponderado	**93.12**
Ensemble Top-3 Melhores	83.74

REFERÊNCIAS

[1] J. Wei *et al.*, "A Petri Dish for Histopathology Image Analysis", International Conference on Artificial Intelligence in Medicine (AIME), 12721:11-24, 2021

[2] "Understanding your pathology report: Colon polyps (sessile or traditional serrated adenomas)" — American Cancer Society, 07 de Jun. 2023, https://www.cancer.org/cancer/diagnosis-staging/tests/biopsy-and-cytology-tests/understanding-your-pathology-report/colon-pathology/colon-polyps-sessile-or-traditional-serrated-adenomas.html (Acesso em 12 de Ago. 2025).
[3] L. N. Smith and N. Topin, Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. 2018. [Online]. Disponível em: https://arxiv.org/abs/1708.07120
[4] "OneCycleLR — PyTorch 2.7 documentation," Pytorch.org, 2024. https://docs.pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.OneCycleLR.html. (Acesso em 16 de Ago. 2025).
[5] F. Baidoun et al., "Colorectal Cancer Epidemiology: Recent trends and Impact on Outcomes.," Current drug targets, 2020, doi: 10.2174/1389450121999201117115717.
[6] K. He, X. Zhang, S. Ren, and J. Sun, Deep Residual Learning for Image Recognition. 2015. [Online]. Disponível em: https://arxiv.org/abs/1512.03385
[7] K. Simonyan and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition. 2015. [Online]. Disponível em: https://arxiv.org/abs/1409.1556
[8] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, Densely Connected Convolutional Networks. 2018. [Online]. Available: https://arxiv.org/abs/1608.06993
[9] S. Mukherjee, "The Annotated ResNet-50 - TDS Archive - Medium," Medium, 18 de Ago. 2022. https://medium.com/data-science/the-annotated-resnet-50-a6c536034758 (Acesso em 16 de Ago. 2025).
[10] Melanie, "Unveiling the Secrets of the VGG Model: A Deep Dive with Daniel," Data Science Courses — DataScientest, 09 de Set. 2023. https://datascientest.com/en/unveiling-the-secrets-of-the-vgg-model-a-deep-dive-with-daniel (Acesso em 16 de Ago. 2025).
[11] A. Muniasamy *et al.*, "Investigating Hybrid Quantum-Assisted Classical and Deep Learning Model for MRI Brain Tumor Classification,"Journal of Image and Graphics, Vol. 13, No. 1, pp. 123-129, 2025
[12] C. S. Sant'Anna, C. A. C. de Albuquerque, S. C. Baraúna, and G. R. de Oliveira Filho, "Prática deliberada no ensino de histologia na graduação em Medicina: estudo prospectivo randomizado e controlado," Revista Brasileira de Educação Médica, vol. 46, p. e082, Jun. 2022, doi: https://doi.org/10.1590/1981-5271v46.2-20210448.
[13] N. R. N. Souza, "História do Microscópio e importância para o desenvolvimento científico," Trabalho de Conclusão de Curso (Graduação em Física – Licenciatura), Centro de Ciências, Universidade Federal do Ceará, Fortaleza, 2023.
[14] J. F. B. de Moura, W. E. S. Eulálio, M. T. B. Silva, e R. R. Bacchi, "Integração de tecnologias digitais para suprir as lacunas no ensino de histopatologia médica", Rev. DELOS, vol. 17, nº 62, p. e2982, dez. 2024.
[15] M. H. Alali, A. Roohi, S. Angizi, and J. S. Deogun, "Enabling Intelligent IoTs for Histopathology Image Analysis Using Convolutional Neural Networks," Micromachines, vol. 13, no. 8, 2022, doi: 10.3390/mi13081364.
[16] T. Araki, "The history of optical microscope," Mechanical Engineering Reviews, vol. 4, no. 1, pp. 16-0024216-00242, 2017, doi: https://doi.org/10.1299/mer.16-00242.
[17] M. Kang, H. Song, S. Park, D. Yoo, and S. Pereira, Benchmarking Self-Supervised Learning on Diverse Pathology Datasets. 2023. [Online]. Disponível em: https://arxiv.org/abs/2212.04690
[18] J. Wu et al., "An automatic classification method of testicular histopathology based on SC-YOLO framework," BioTechniques, vol. 76, no. 9, pp. 443–452, Sep. 2024, doi: https://doi.org/10.1080/07366205.2024.2393544.