

Predição da Presença de Carcinoma Ductal Invasivo em Imagens de Histologia de Câncer de Mama Utilizando Aprendizado Profundo

Pedro Henrique Campos Lorenzetti¹, Murilo Falleiros Lemos Schmitt¹

¹Departamento de Ciência da Computação (DECOMP)
Universidade Estadual do Centro-Oeste (UNICENTRO)
Caixa Postal 730 – 85.040-167 – Guarapuava – PR – Brasil

phlorenzetti@gmail.com, muriloschmitt@gmail.com

Abstract. *The analysis of invasive ductal carcinoma images in breast cancer is crucial for early diagnosis and effective treatment. This study applies Deep Learning algorithms, specifically Convolutional Neural Networks, to classify histological images for the presence of the disease. The Deep Learning models demonstrated superior performance compared to traditional algorithms such as random forest, XGBoost, and K-nearest neighbors, achieving over 91% accuracy on the test dataset. The work highlights the importance of neural networks in reducing diagnostic variability and supporting the work of pathologists.*

Resumo. *A análise de imagens de carcinoma ductal invasivo em câncer de mama é fundamental para o diagnóstico precoce e tratamento eficaz. Este estudo aplica algoritmos de Aprendizado Profundo, especificamente Redes Neurais Convolucionais, para classificar imagens histológicas quanto à presença da doença. Os modelos de Aprendizado Profundo mostraram desempenho superior em relação aos algoritmos tradicionais como FA, XGBoost e K-vizinhos mais próximos, alcançando acurácia acima de 91% na base de testes. O trabalho destaca a importância das redes neurais na redução da variabilidade diagnóstica e no suporte ao trabalho dos patologistas.*

1. Introdução

Segundo o Instituto Nacional do Câncer¹, a estimativa de 2023 a 2025 aponta que serão diagnosticados ao todo 704 mil casos de câncer no Brasil, dado que as neoplasias representam a segunda principal causa de morte prematura em todo o mundo. Quanto ao câncer de mama, este é o principal acometimento histológico feminino no Brasil, representando um total de 41,89 casos a cada 100.000 mulheres, sendo o Carcinoma Ductal Invasivo (IDC, do inglês *Invasive Ductal Carcinoma*) o tipo mais comum, compreendendo 80% a 90% do total de casos [Santos et al. 2023]. Tratando-se desse subtipo de câncer, a fase do estadiamento neoplásico implica no prognóstico ou diagnóstico precoce, e consequentemente na eficácia do tratamento, tendo em vista que essa é mais presente em casos de tumores pequenos e sem comprometimento da rede linfática [Gennaro et al. 2022].

A análise de imagens de IDC, quando feita de forma manual, exige um processo trabalhoso e muitas vezes suscetível de alterações nos dados que prejudicam os

¹Instituto Nacional do Câncer (INCA), Brasil. Disponível em: <https://www.inca.gov.br/>. Acesso em: 16 de julho de 2024

resultados. Em contrapartida, está se tornando cada vez mais comum abordagens algorítmicas para solucionar esse problema, pois além de se mostrarem benéficas na maioria das aplicações, diminuindo o tempo de adquirir informações precisas sobre as imagens, também podem ajudar a reduzir a variabilidade da interpretação entre patologistas [Janowczyk and Madabhushi 2016].

O processo de prognóstico e diagnóstico pode ser facilitado por meio de técnicas de Aprendizado Profundo (AP), que consiste no aprendizado de computadores através da experiência em termos de conceitos hierarquizados, com cada conceito definido por meio da sua relação com conceitos mais simples, sendo uma ferramenta útil na tarefa de classificação de imagens [Goodfellow et al. 2016].

Nesse sentido, este trabalho pretende estudar e aplicar algoritmos de AP em imagens de histologia de câncer de mama, a fim de classificar quais imagens apresentam o *IDC* e quais não apresentam. Além disso, visa comparar esses algoritmos com outros de Aprendizado de Máquina (AM). A pesquisa inclui a implementação de modelos de Redes Neurais Convolucionais (*CNN*, do inglês *Convolutional Neural Network*), adaptados para a tarefa específica de detecção de câncer de mama. Esses modelos foram comparados com algoritmos tradicionais, utilizando uma base de dados de histopatologia de mama disponível publicamente. A comparação objetiva identificar o modelo que oferece melhor desempenho em termos de precisão, *recall*, *F1-score* e matriz de confusão.

O restante do trabalho está organizado da seguinte forma. A Seção 2 apresenta a metodologia aplicada. A Seção 3 discute os trabalhos relacionados. A Seção 4 detalha o desenvolvimento do projeto. A Seção 5 apresenta os experimentos e resultados. A Seção 6 conclui este trabalho e discute trabalhos futuros.

2. Fundamentação Teórica

Esta seção é dedicada para apresentação dos conceitos de *IDC*, Aprendizado de Máquina e Redes Neurais Convolucionais, utilizados no desenvolvimento do trabalho.

2.1. Carcinoma Ductal Invasivo

Atualmente o Câncer de Mama é a doença maligna que mais afeta a população feminina. Um dos principais subtipos do cancro mamário é o *IDC* que se caracteriza por um arranjo irregular de células, que se apresentam em formato de ninho ou massa dentro do tecido mamário [Zhao et al. 2022]. Essa massa pode ser ovalada, lobulada ou irregular, com contorno indefinido ou espiculado, ocorrendo na forma multicêntrica e bilateral [Gennaro et al. 2022].

Ao realizar a leitura de exames histológicos de imagem, existem alterações sutis entre condições mamárias benignas e malignas. Em alterações benignas, como na doença fibrocística, as células são formadas por nódulos delimitados, com os tecidos circundantes sem evidências de invasão celular, enquanto nas alterações malignas, como é o caso do *IDC*, o crescimento das células é geralmente irregular, exibindo estruturas ductais anormais e uma desorganização do tecido. Tais alterações podem passar despercebidas ao olho humano, o que torna o uso do AP essencial para a identificação da patologia, promovendo o diagnóstico precoce [Moreira 2023].

A Figura 1 apresenta uma imagem histológica de tecido mamário com doença fibrocística, mostrando células apócrinas que são células glandulares modificadas e áreas

císticas, caracterizados por pequenos cistos cheios de líquido. Já na Figura 2 tem-se uma imagem histológica de tecido mamário representativo de *IDC*, onde é possível observar as células neoplásicas (células cancerosas anormais), uma desorganização arquitetural típica desta patologia, e os ductos mamários que são os canais que transportam leite, preenchidos por células neoplásicas [Moreira 2023].

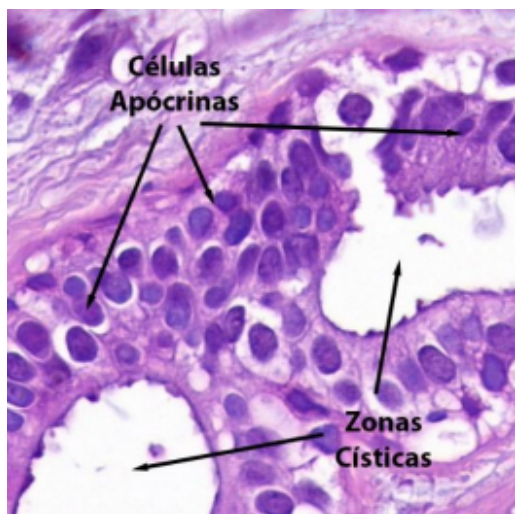


Figura 1. Doença Fibrocística.
Fonte: [Moreira 2023].

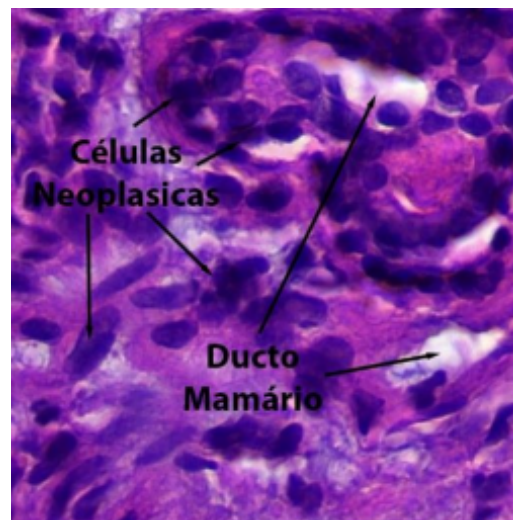


Figura 2. Carcinoma ductal invasivo. Fonte: [Moreira 2023].

2.2. Aprendizado de Máquina

Os algoritmos de AM estão revolucionando a medicina, melhorando diagnósticos, prognósticos e a descoberta de novos tratamentos. Ao analisar grandes volumes de dados, eles identificam padrões sutis em imagens médicas e registros de saúde, permitindo diagnósticos mais precisos e decisões clínicas informadas. Esses algoritmos também podem prever o curso de doenças, personalizar tratamentos e acelerar a pesquisa de novos medicamentos [Garzon and Barbosa 2023].

Quando é construído um programa que usa exemplos para melhorar seu desempenho, está sendo utilizado AM. Essa técnica é baseada em um projeto de código orientado a dados, ou seja, por meio de um grande volume de dados, aprendem automaticamente e geram hipóteses. Nessa técnica quanto melhor a qualidade dos dados, mais precisa é a síntese feita pela máquina. O AM é dividido em três principais métodos, o supervisionado, não supervisionado e por reforço [Ludermir 2021]. Neste trabalho será utilizado a primeira.

No aprendizado supervisionado, cada exemplo fornecido ao algoritmo vem com a resposta correta, ou seja, um rótulo indicando a classe do exemplo. Em um problema de classificação de imagens, os rótulos indicam se uma imagem é de *IDC* positivo ou negativo. Cada exemplo é representado por um conjunto de atributos e seu respectivo rótulo. Logo o objetivo do algoritmo é aprender a classificar corretamente novos exemplos que ainda não foram rotulados. Quando os rótulos são categorias discretas, isto é chamado de classificação. Este é o método de AM mais amplamente utilizado [Ludermir 2021].

2.3. Redes Neurais Artificiais

Um bom exemplo para entender como o AM funciona é por uma de suas principais aplicações, as Redes Neurais, que são chamadas de redes porque são representadas pela composição de várias funções, formando um grafo acíclico direcionado. Por exemplo, podem-se conectar várias funções em uma cadeia para formar uma sequência de operações, essas cadeias determinam a profundidade do modelo [Goodfellow et al. 2016].

As primeiras camadas são as de entrada, elas recebem os dados brutos e transformam as características de entrada, transmitindo-as para as camadas subsequentes. As camadas ocultas são intermediárias entre a entrada e a saída, e realizam cálculos não visíveis diretamente para o usuário. Elas transformam os dados de entrada, multiplicando eles por pesos, e após isto, passando-os por funções de ativação, que introduzem a não-linearidade ao modelo permitindo a rede aprender representações mais complexas dos dados de entrada [Aggarwal 2018]. As últimas camadas são as de saída, e o treinamento da rede ajusta essa cadeia de funções para que a saída da rede se aproxime da saída desejada, utilizando exemplos fornecidos pelos dados de treinamento [Goodfellow et al. 2016].

Essas redes são chamadas de neurais porque são inspiradas pela neurociência, onde cada camada oculta é composta por várias unidades que atuam em paralelo, semelhante a neurônios. A escolha das funções usadas nas camadas é orientada por observações neurocientíficas, mas a pesquisa moderna em redes neurais é guiada por disciplinas matemáticas e de engenharia. Portanto, as redes neurais devem ser vistas como máquinas de aproximação de funções, projetadas para generalização estatística, e não como modelos precisos da função cerebral [Goodfellow et al. 2016].

2.4. Aprendizado Profundo

O AM e suas aplicações conseguem resolver muitos problemas, mas quando está sendo usado um conjunto de dados brutos onde a extração de características é de extrema dificuldade, o reconhecimento se torna complexo, por motivo de estar se tratando de elementos cujos fatores precisam de uma sofisticada análise para serem identificados, e para isso o AP consegue ajudar de maneira assertiva, afinal esta técnica permite que se consiga fazer representações complexas utilizando conceitos mais simples [Goodfellow et al. 2016].

Existem diversos tipos de redes neurais que fazem uso da técnica de AP. Dentre esses, existem as *CNNs*, que são projetadas para trabalhar com dados estruturados em matriz, capturando padrões locais por meio de operações de convolução [Aggarwal 2018]. Por conta deste fato a *CNN* tem mostrado eficiência nas tarefas de classificação de imagem.

2.5. Redes Neurais Convolucionais

As *CNNs* são projetos de AP construídos para processar dados que se apresentam no formato matricial, um exemplo são imagens coloridas, representadas por três arranjos bidimensionais de intensidade de *pixels*, onde cada um deles condiz com um dos três canais de cores, vermelho, verde e azul. A *CNN* apresenta uma arquitetura padrão que envolve diversas camadas, que podem ser camadas convolucionais, de *pooling* ou densas. Elas são organizadas em grades, mantendo relacionamentos herdados de uma camada para a próxima. Cada uma delas possui altura, largura e profundidade que fazem referência ao número de canais de cor primária na imagem de entrada ou no mapa de recursos nas

camadas ocultas. Juntas elas detectam padrões locais e reduzem a dimensionalidade da representação, percebendo pequenos deslocamentos e distorções que vão influenciar na classificação [LeCun et al. 2015].

2.5.1. Camada Convolutiva

A camada convolutiva utiliza-se da operação de convolução que envolve colocar o filtro em cada posição da imagem e realizar um produto entre os parâmetros do filtro e a região correspondente da imagem de entrada como visto na Figura 3. Esse filtro é uma pequena janela quadrada que passa sobre a imagem, analisando partes dela para detectar características específicas, como bordas ou texturas. Ele é menor que a imagem à qual está sendo aplicado. Cada posição em que este filtro será aplicado define um *pixel* espacial na próxima camada, com o número de posições possíveis determinando a altura e a largura da próxima camada oculta. A profundidade da camada que vem a seguir é determinada pelo número de filtros aplicados na camada atual, resultando em um mapa de características que captura diferentes detalhes da imagem, desde os mais simples aos mais complexos [Aggarwal 2018].

$$\begin{array}{|c|c|c|} \hline \text{Entrada} & & \\ \hline 0 & 1 & 2 \\ \hline 3 & 4 & 5 \\ \hline 6 & 7 & 8 \\ \hline \end{array} * \begin{array}{|c|c|} \hline \text{Filtro} & \\ \hline 0 & 1 \\ \hline 2 & 3 \\ \hline \end{array} = \begin{array}{|c|c|} \hline \text{Saída} & \\ \hline 19 & 25 \\ \hline 37 & 43 \\ \hline \end{array}$$

Figura 3. Operação de Convolução. Adaptado de [Zhang et al. 2023]

Entre as camadas são utilizadas funções de ativação não-lineares. Neste trabalho será utilizada a função *ReLU* para ligar as camadas convolucionais. Ela é uma função de ativação que retorna o valor de entrada diretamente se este for positivo, caso contrário retorna zero [Agarap 2018].

2.5.2. Camada de Pooling

Esta camada abrange uma técnica muito importante para a *CNN*, afinal, ela visa reduzir gradualmente a resolução espacial das representações ocultas. Isso é feito agregando informações ao longo das camadas, aumentando assim o campo receptivo de cada nó oculto. Esse processo é essencial para tarefas globais, como a detecção de objetos em uma imagem inteira, pois ajuda a criar representações globais enquanto mantém as vantagens das camadas convolucionais intermediárias [Zhang et al. 2023]. As camadas de *pooling* servem a dois propósitos principais, reduzir a sensibilidade das camadas convolucionais à tradução da imagem, o que é importante, pois objetos raramente aparecem na mesma posição exata em diferentes imagens, e reduzir a resolução espacial das representações ocultas, permitindo uma representação mais condensada e eficiente dos dados.

Existem dois tipos mais comuns de *poolings* utilizados em *CNNs*, sendo esses [Zhang et al. 2023]:

- **Pooling Máximo:** seleciona o valor máximo em cada janela de *pooling*.
- **Pooling Médio:** calcula a média dos valores na janela de *pooling*.

Neste trabalho foi utilizado o *pooling* máximo, porque ele reduz a resolução espacial das imagens, realçando as características mais proeminentes e importantes de uma imagem, eliminando informações menos expressivas [Aggarwal 2018]. Isto influencia em tarefas de detecção e classificação, onde as características mais marcantes são as mais relevantes.

2.5.3. Camadas Densas

As camadas densas são responsáveis por conectar todos os neurônios de uma camada a todos os neurônios da próxima camada, permitindo uma integração completa da informação. No caso da *CNN*, elas são posicionadas após as camadas de convolução e *pooling*, recebendo a entrada na forma de um vetor de características extraídas anteriormente. Essas camadas transformam características em uma forma que pode ser utilizada para a classificação final ou outras tarefas específicas. Em suma a importância da camada densa reside no fato que elas contêm a maioria dos parâmetros treináveis da rede, o que lhes permite aprender representações complexas e discriminar de forma eficaz entre diferentes classes [Basha et al. 2020].

2.5.4. Dropout

Dropout seleciona uma fração de neurônios para serem desativados aleatoriamente ao zerar seus sinais de avanço durante cada iteração da fase de treinamento. Isso evita *overfitting*, que é quando as unidades neurais se adaptam excessivamente aos dados de treinamento, e não generalizam aos dados de teste. Logo, esta técnica permite melhor generalização em dados de teste não vistos [Srivastava et al. 2014].

2.6. Métricas de Avaliação

Para avaliar algoritmos de AP em problemas de classificação geralmente são utilizadas algumas métricas de avaliação. Neste trabalho, assim como na literatura [Yacoubi and Axman 2020], Acurácia, Precisão, *Recall* e *F1-Score* serão as métricas de avaliação utilizadas no modelo:

- **Acurácia:** mede a proporção de previsões corretas entre o total de previsões feitas.
- **Precisão:** é usada para medir a proporção de verdadeiros positivos entre os exemplos classificados como positivos pelo modelo.
- **Recall:** mede a proporção de verdadeiro positivos entre todos os exemplos que realmente são positivos.
- **F1-score:** é a média harmônica da precisão e do *recall*.
- **Matriz de Confusão:** é uma tabela que permite a visualização do desempenho do algoritmo de classificação, mostrando o número de verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos.

3. Trabalhos Relacionados

Esta seção será dedicada aos trabalhos que são correlatos ao tema deste projeto. Os artigos usados foram extraídos em bases da IEEE, *Google Scholar* e *Science Direct*. Foram filtrados artigos publicados após 2020 usando palavras-chave como “*cnn*” e “*deep learning*” na busca de textos que usem das tecnologias implementadas na pesquisa, “*kaggle*”² tentando encontrar trabalhos que usaram esse site para retirar base de dados de AM, “*breast cancer*” e “*idc positive negative*” para encontrar trabalhos relacionados ao problema. Após esse processo foram selecionados 4 artigos para discussão. A seguir, são discutidas algumas abordagens dos trabalhos escolhidos, destacando seus objetivos, métodos e contribuições principais.

O trabalho de [Chapala and Sujatha 2020] utiliza uma Rede Neural Convolutio-nal para extrair características e treinar um modelo capaz de classificar imagens como malignas ou benignas. Esta abordagem visa melhorar a precisão e eficiência na detecção de IDC, facilitando o trabalho dos patologistas e proporcionando suporte no diagnóstico.

Outra proposta envolve a implementação de um modelo de CNN para a classificação do IDC, equilibrando o conjunto de dados e ajustando hiper-parâmetros para superar modelos pré-treinados [Karatayev et al. 2021].

Além disso, [Das et al. 2021] propõe um modelo de AP para detectar IDC, que por meio de redes convolucionais, atinge acima de 98% em métricas de avaliação como acurácia, precisão, *recall* e *F1-score*, ela destaca-se por quantificar a incerteza nas previsões, melhorando a confiabilidade e a precisão dos diagnósticos automatizados.

Por fim, uma abordagem chamada Rede Neural Convolutacional Bayesiana Otimizada (OBCNN) foi proposta por [Ezzat and Hassanien 2023], para a detecção automatizada de IDC. Esta técnica se destaca por quantificar e inferir a incerteza nas previsões, melhorando a capacidade de diagnóstico e aumentando a confiabilidade das decisões automatizadas.

4. Desenvolvimento

Com o propósito de desenvolver o trabalho, técnicas de AP e AM foram aplicadas por meio de ferramentas e base de dados. Esta seção abordará como este processo foi realizado.

4.1. Base de Dados

A base de dados escolhida para o desenvolvimento do trabalho foi “*Breast Histopathology Images*” [Mooney 2024] retirada da plataforma de ciência de dados *Kaggle*. A escolha dessa base de dados foi baseada em seu extenso uso dentro da área acadêmica, sendo aproveitada por diversos artigos de inteligência artificial.

Originalmente esse conjunto de dados se baseia em 162 imagens de câncer de mama digitalizadas e aumentadas 40 vezes em relação ao tamanho real das imagens, dessas foram extraídos 277.524 recortes de imagem com o tamanho de 50 *pixels* de altura e 50 *pixels* de largura, assim obtendo 198.738 negativos para IDC e 78.786 positivos para IDC. O formato do arquivo de cada recorte é por exemplo `u_xX_yY_classC.png`

²<https://www.kaggle.com/>

onde u é o ID do paciente, X é a coordenada x de onde este recorte foi extraído, Y é a coordenada y de onde este recorte foi extraído e C indica a classe, onde 0 é não-*IDC* e 1 é *IDC*. Um exemplo de um arquivo para o caso negativo da doença e um exemplo de recorte para o caso positivo da doença, são mostrados respectivamente nas Figuras 4 e 5.

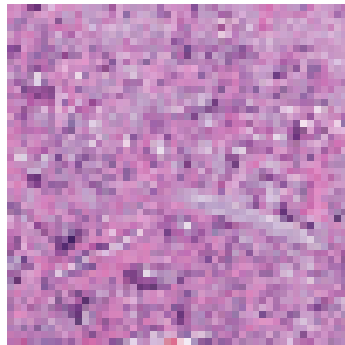


Figura 4. Caso negativo para IDC. Fonte: [Mooney 2024]

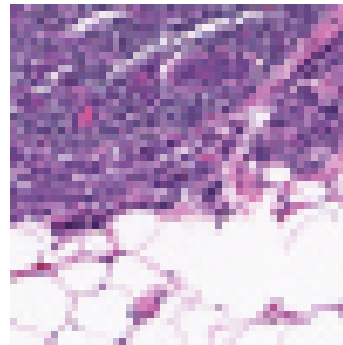


Figura 5. Caso positivo para IDC. Fonte: [Mooney 2024]

4.2. Tratamento de Dados

O processo de tratamento dos dados envolve diversas etapas essenciais, que garantem que as imagens sejam devidamente separadas em conjuntos de treino, validação e teste. Isto impede que dados de uma etapa do algoritmo vazem para outra. Também transformando as imagens para que os modelos tenham mais exemplos das mesmas durante o treinamento.

4.2.1. Divisão dos Dados em Treino, Validação e Teste

Em um primeiro momento os dados foram divididos em dois subconjuntos denominados treino e teste, alocando 80% dos dados para o conjunto de treino e 20% para o conjunto de teste. Após esta divisão os dados foram novamente separados, sendo assim, 80% dos dados do conjunto treino original foram alocados para o novo conjunto de treino, e 20% foram destinados ao conjunto validação. Os tamanhos resultantes dos conjuntos de dados foram os seguintes:

- Conjunto de Treino: 177.615 amostras (80% dos 80% iniciais, ou seja, 64% do total)
- Conjunto de Validação: 44.404 amostras (20% dos 80% iniciais, ou seja, 16% do total)
- Conjunto de Teste: 55.505 amostras (20% do total)

4.2.2. Transformações

Para aumentar a robustez do modelo e melhorar a generalização, foram definidas diversas transformações (*data augmentation*) para os conjuntos de treino. As transformações são

fundamentais para melhorar a performance do modelo em dados não vistos e reduzir o risco de *overfitting* [Aggarwal 2018].

As transformações aplicadas ao conjunto de treino têm como objetivo aumentar a variabilidade dos dados, permitindo que o modelo seja mais robusto a diferentes variações que possam ocorrer em imagens reais. Nesse processo de transformações as imagens tem suas bordas preservadas para que a informação delas não seja perdida, também são giradas horizontalmente e verticalmente, por fim são rotacionadas em até 20 graus, após isso elas são normalizadas e padronizadas.

4.3. Modelo

A arquitetura de AP utilizada nesse trabalho é uma adaptação do VGG [Simonyan and Zisserman 2015]. Este modelo de *CNN* é caracterizado pelo uso de algumas camadas convolucionais seguidas de uma camada de *pooling* máximo, aplicadas em toda a profundidade da rede. Essas camadas são seguidas por 3 camadas densas ao final da rede. A VGG pode ter entre 11 e 19 camadas de peso, dependendo de sua configuração. Serão explorados dois modelos, a VGG16 que corresponde a 16 camadas de peso e a VGG19 que apresenta 19 camadas de peso. Os pesos dos modelos são pré-treinados com a base de dados *ImageNet*³ que contém milhões de imagens rotuladas em mil categorias diferentes, a fim de acelerar o processo de treinamento e levar a melhores resultados em menos tempo [Deng et al. 2009].

4.3.1. VGG16

A VGG16 é organizada em 8 blocos distintos, cada um com funções específicas, como mostra a Figura 6:

³<https://www.image-net.org/>

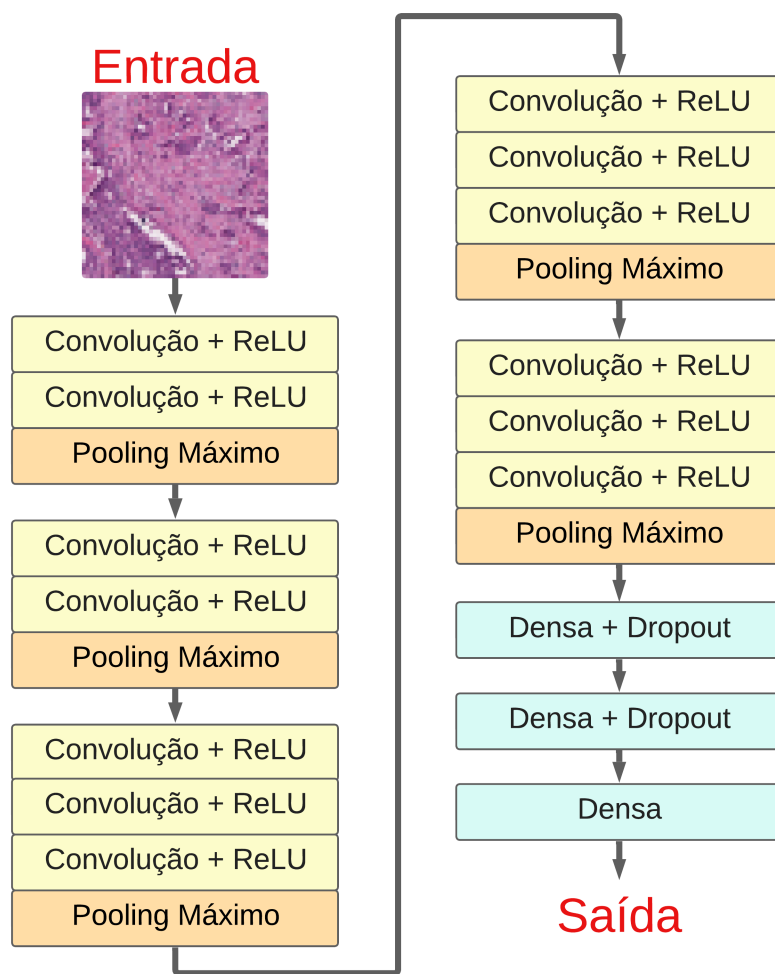


Figura 6. Arquitetura VGG16.

Blocos de Convolução e *Pooling*:

- 1º e 2º bloco: Cada um contém 2 camadas de convolução seguidas por uma camada de *pooling*. No 1º bloco as duas camadas possuem 64 filtros de tamanho 3x3 e no 2º bloco as duas camadas dispõem de 128 filtros de 3x3.
- 3º ao 5º bloco: Cada um possui 3 camadas de convolução seguidas por uma camada de *pooling*. No 3º bloco as três camadas têm 256 filtros 3x3, já no 4º bloco as camadas contam com 512 filtros 3x3, por fim no 5º bloco as camadas também reúnem 512 filtros 3x3.

Blocos Densos:

- 6º e 7º bloco: Cada um contém uma camada densa que usa o *dropout*. As duas camadas densas carregam 4096 neurônios.
- 8º bloco: Contém uma camada densa final que produz a saída da rede. Ela traz 1000 neurônios.

4.3.2. VGG19

Nos mesmos moldes da VGG16, a VGG 19 também é organizada em 8 blocos distintos, porém ela apresenta uma camada convolucional a mais do 3º ao 5º bloco.

Blocos de Convolução e *Pooling*:

- 1º e 2º bloco: Cada um contém 2 camadas de convolução seguidas por uma camada de *pooling*. Para as camadas do 1º bloco, são 64 filtros de tamanho 3x3 e para as camadas do 2º bloco, são 128 filtros 3x3.
- 3º ao 5º bloco: Cada um possui 4 camadas de convolução seguidas por uma camada de *pooling*. O 3º bloco as camadas apresentam 256 filtros 3x3, já para o 4º e 5º bloco, em todas as camadas, são 512 filtros 3x3.

Blocos Densos:

- 6º e 7º bloco: Cada um contém uma camada densa que usa o *dropout*. Assim como na VGG16, nesta, cada camada densa possui 4096 neurônios.
- 8º bloco: Contém uma camada densa final que produz a saída da rede. Ela contém 1000 neurônios.

4.4. Comparação de Algoritmos

Em artigos relacionados, os modelos principais de *CNNs* utilizados são geralmente comparados com outros algoritmos que não são AP. Como feito na literatura, neste trabalho são avaliados os algoritmos, Floresta Aleatória (FA), *XGBoost* e K-vizinhos mais próximos (*KNN*, do inglês *K-nearest neighbors*) [Sajid et al. 2023], a fim de comparar com a VGG.

4.4.1. Floresta Aleatória

É uma técnica de AM que desenvolve múltiplas árvores de decisão usadas para classificação e regressão. Uma árvore de decisão consiste em ramos e nós, onde cada nó divide o subconjunto de características de acordo com a regra de decisão associada [Sajid et al. 2023].

A floresta aleatória implementada neste trabalho realiza a classificação de imagens com base em características extraídas de histogramas de intensidade e cor. As imagens são redimensionadas para 50x50 *pixels* e convertidas para escala de cinza, após isso são calculados histogramas de intensidade para a imagem em cinza e para cada canal de cor (azul, verde e vermelho), essas características são extraídas para os conjuntos de treinamento e validação, e um classificador de FA é treinado com essas características e seus rótulos correspondentes.

4.4.2. *XGBoost*

É outro algoritmo que tem seu modelo baseado em árvores. Nele, novas árvores de decisão são adicionadas sequencialmente para corrigir os erros das árvores anteriores. Em

seguida, ele minimiza a perda de cada árvore usando um algoritmo de descida de gradiente. A função objetivo deste modelo considera tanto a precisão das previsões quanto a complexidade das árvores, ajudando a evitar o *overfitting* ao penalizar árvores mais complexas e pesos extremos [Sajid et al. 2023].

Neste trabalho o *XGBoost* aplicado utiliza as imagens de tecido mamário convertidas em vetores unidimensionais e usadas como entrada do modelo, então, durante o treinamento o *XGBoost* ajusta iterativamente as previsões das árvores, minimizando os erros das iterações anteriores através da descida de gradiente.

4.4.3. K-Vizinhos Mais Próximos

O algoritmo *KNN* classifica uma nova instância comparando-a com outras instâncias já conhecidas, identificando as mais próximas com base em uma métrica de distância. Desse jeito a classificação é feita atribuindo-se à nova instância a classe mais comum entre seus vizinhos mais próximos, sendo uma técnica amplamente utilizada para o reconhecimento de padrões [Sajid et al. 2023].

O *KNN* utilizado armazena os exemplos de treinamento e para classificar uma nova instância, calcula a distância entre a nova instância e todos os exemplos armazenados, selecionando os 3 vizinhos mais próximos. A classe da nova instância é determinada pela votação majoritária entre os rótulos dos vizinhos mais próximos. No projeto, imagens de tecidos mamários são transformadas e achatadas, permitindo que o *KNN* classifique novas imagens com base nas características extraídas dos exemplos de treinamento.

4.5. Metodologia

A metodologia para este trabalho envolveu o treinamento dos modelos VGG16 e VGG19 em várias épocas. Durante cada época, os dados de entrada foram processados pelo modelo, e a saída gerada foi comparada com os rótulos verdadeiros utilizando uma função de perda para calcular o erro. Esse erro foi então utilizado para ajustar os pesos do modelo através da retro-propagação e do uso de um otimizador.

Foram utilizados dois tipos de otimizadores, o SGD⁴ que atualiza os pesos do modelo com base em uma média dos gradientes calculados a partir de um mini-lote de dados, e o Adam⁵, que por outro lado, combina as vantagens do SGD com a adaptação das taxas de aprendizado individuais para cada parâmetro, acelerando o processo de convergência. A taxa de aprendizado é um parâmetro importante que determina o quanto os pesos do modelo são ajustados durante o treinamento, tornando de suma importância ajustá-la ao longo do tempo para evitar que o modelo fique preso em soluções ruins, melhorando a eficiência do aprendizado [Smith 2017]. A acurácia do treinamento foi monitorada ao longo do processo, calculando-se a proporção de previsões corretas. Após cada época de treinamento, o modelo foi avaliado utilizando um conjunto de validação, onde a perda e a acurácia foram novamente calculadas. Os pesos do modelo que resultaram na maior acurácia de validação foram salvos como os melhores.

⁴<https://pytorch.org/docs/stable/generated/torch.optim.SGD.html#torch.optim.SGD>

⁵<https://pytorch.org/docs/stable/generated/torch.optim.Adam.html#torch.optim.Adam>

4.6. Bibliotecas e Ferramentas

Para o desenvolvimento do trabalho, as principais bibliotecas e ferramentas utilizadas:

- **Pytorch**⁶: uma biblioteca de AM utilizada neste projeto para implementação de arquiteturas de AP.
- **Scikit-learn**⁷: esta biblioteca foi utilizada para importar e utilizar as métricas de avaliação do modelo como Acurácia, Precisão, *Recall* e *F1 Score* e Matriz de Confusão. Também foi aplicada na implementação de algoritmos como o K-Vizinhos mais próximos e a Floresta Aleatória.
- **XGB**⁸: foi utilizada na implementação do algoritmo de *XGBoost*.
- **Google Colab**⁹: todos os algoritmos desenvolvidos e apresentados neste trabalho foram testados na plataforma *Google Colab*.

5. Experimentos e Resultados

Esta seção discute os resultados obtidos pelos experimentos dos modelos de *CNN* e os demais algoritmos aplicados neste trabalho.

5.1. Ajuste de Hiper-parâmetros

O ajuste de hiper-parâmetros foi realizado utilizando-se os conjuntos de dados de treino e validação, com o objetivo de otimizar o desempenho dos modelos VGG16 e VGG19. Os principais hiper-parâmetros ajustados foram a taxa de aprendizado e o otimizador, sendo testado tanto o SGD quanto o Adam.

A Tabela 1 mostra os resultados para VGG16 usando o otimizador SGD, e logo após, na Tabela 2 usando o otimizador Adam.

Tabela 1. Resultados dos experimentos com diferentes taxas de aprendizado usando o otimizador SGD na VGG16

Experimento	Taxa de Aprendizado	Acurácia da Validação	Perda da Validação
1	0,00001	0,8670	0,3154
2	0,0001	0,8905	0,2603
3	0,001	0,9076	0,2235
4	0,01	0,9109	0,2197

Tabela 2. Resultados dos experimentos com diferentes taxas de aprendizado usando o otimizador Adam na VGG16

Experimento	Taxa de Aprendizado	Acurácia da Validação	Perda da Validação
1	0,000001	0,8898	0,2603
2	0,00001	0,9112	0,2184
3	0,0001	0,9164	0,2054
4	0,001	0,7161	0,5966

⁶<https://pytorch.org/>

⁷<https://scikit-learn.org/stable/>

⁸<https://xgboost.readthedocs.io/en/stable/>

⁹Google Colaboratory - Ferramenta para desenvolvimento de projetos em Python

Pode-se observar, por meio dos experimentos relatados na Tabela 1 e Tabela 2, que para o modelo VGG16, a melhor configuração é o otimizador Adam e a taxa de aprendizado com valor de 0,0001. Esta configuração permite o modelo atingir uma acurácia de 0,9164 e uma perda de 0,2054 na base de validação, apresentando a maior acurácia e a menor perda entre os experimentos.

A Tabela 3 apresenta os resultados obtidos para a VGG19 com o uso do otimizador SGD, enquanto a Tabela 4 mostra os resultados utilizando o otimizador Adam.

Tabela 3. Resultados dos experimentos com diferentes taxas de aprendizado usando o otimizador SGD na VGG19

Experimento	Taxa de Aprendizado	Acurácia da Validação	Perda da Validação
1	0,00001	0,8650	0,3162
2	0,0001	0,8914	0,2531
3	0,001	0,9045	0,2267
4	0,01	0,9089	0,2222

Tabela 4. Resultados dos experimentos com diferentes taxas de aprendizado usando o otimizador Adam na VGG19

Experimento	Taxa de Aprendizado	Acurácia da Validação	Perda da Validação
1	0,000001	0,8861	0,2671
2	0,00001	0,9078	0,2265
3	0,0001	0,9139	0,2118
4	0,001	0,7161	0,5966

As Tabelas 3 e 4 apresentam que a melhor configuração para a VGG19, é o otimizador Adam e a taxa de aprendizado com o valor de 0,0001. A configuração mostrada permite o modelo atingir uma acurácia de 0,9139 e uma perda de 0,2118 na base de validação, apresentando a maior acurácia e a menor perda entre os experimentos.

Por meio dos experimentos observa-se que o otimizador Adam obteve desempenhos melhores em relação ao SGD no que diz respeito a acurácia. Isto é explicado devido a capacidade do Adam de adaptar as taxas de aprendizado para cada parâmetro individualmente. Logo a taxa de aprendizado 0,0001, mesclada com o otimizador Adam, controlou a atualização de pesos de maneira mais estável durante o treinamento, fazendo adaptações de forma com que o aprendizado fosse mais eficiente e a acurácia maior na validação. Em contrapartida o otimizador SGD precisou de taxas de aprendizados maiores para atingir resultados mais robustos.

Após os resultados dos experimentos na base de validação, foi selecionado o experimento 3 da Tabela 2, com taxa de aprendizado 0,0001 e que faz uso do otimizador Adam, para representar a VGG16 na base de teste. O experimento 3 da Tabela 4, que apresenta a taxa de aprendizado 0,0001 e utiliza o otimizador Adam, foi escolhido para representar a VGG19 na base de teste. Estes experimentos com estes hiper-parâmetros foram escolhidos, afinal, eles mostraram os melhores resultados em relação a acurácia.

5.2. Resultados Gerais e Discussão

Na Tabela 5, tem-se os resultados dos experimentos de todos os algoritmos na base de testes aplicados no trabalho, usando as métricas de avaliação como parâmetro:

Tabela 5. Resultados dos Testes dos Algoritmos				
Algoritmo	Acurácia	Precisão	Recall	F1-score
VGG16	0,91	0,92	0,92	0,92
VGG19	0,91	0,92	0,92	0,92
FA	0,87	0,87	0,87	0,87
KNN	0,79	0,80	0,79	0,79
XGBoost	0,85	0,85	0,85	0,85

Os resultados dos testes, sugerem que os modelos de *CNN*, VGG16 e VGG19, obtiveram o melhor desempenho, em todas as métricas de avaliação propostas, superando modelos como FA, *KNN* e *XGBoost* na tarefa de classificação da presença de *IDC* em imagens de câncer mama. Isto sugere que quando utiliza-se modelos de *CNN* para esta tarefa, o desempenho tende a ser melhor.

Tal desempenho superior da VGG16 e VGG19, ocorre pelo fato de que as camadas convolucionais, tanto quanto as camadas de *pooling* e camadas densas, quando unidas, permitem que os modelos capturem características complexas das imagens, de forma automática. Já em algoritmos de AM, as características precisam ser extraídas anteriormente, por algum processo, como por exemplo o algoritmo FA, onde é utilizado histogramas de cor para esta função.

Como limitação do trabalho, a diferença da eficácia de algoritmos de AP em relação aos algoritmos de comparação, também pode ser dada pelo fato dos modelos de *CNN* terem sido melhores aprofundados durante o desenvolvimento da pesquisa, isto é, obtiveram um foco especial durante a implementação, tornando a comparação desfavorável para a FA, *KNN* e *XGBoost*, que foram implementados de formas mais básicas em relação ao VGG16 e VGG19.

A matriz de confusão para VGG16 aplicada na base de teste, é representada pela Figura 7. Nela, pode-se observar que entre 39748 imagens sem *IDC* o modelo classifica 37136 imagens corretamente, representando 93,42% de acerto em classificar imagens sem *IDC*. Já para imagens com *IDC*, que são 15757, foram classificadas corretamente, 12733 imagens, representando um acerto de 86,88% em classificar imagens com *IDC*.

Para a VGG19 tem-se a matriz de confusão na Figura 8. Nela, nota-se que de 39748 imagens sem *IDC* o modelo classifica corretamente 38146, representando 95,96% de acerto em classificar imagens sem *IDC*. Para as 15757 imagens com *IDC* o modelo acerta a classificação de 12733 imagens, classificando 80,8% das imagens corretamente.

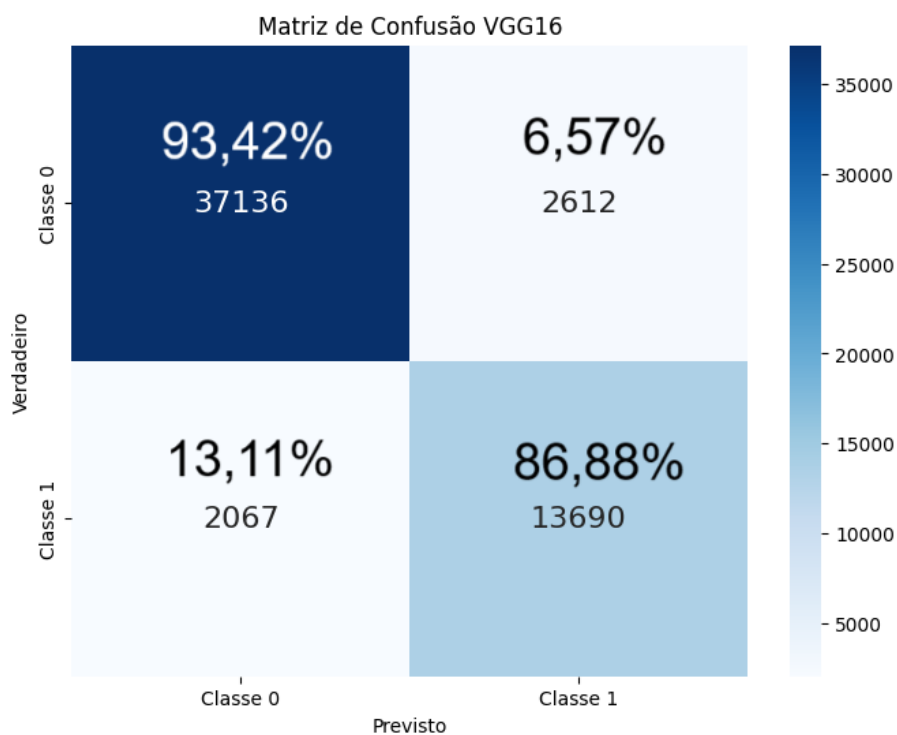


Figura 7. Matriz de Confusão VGG16.

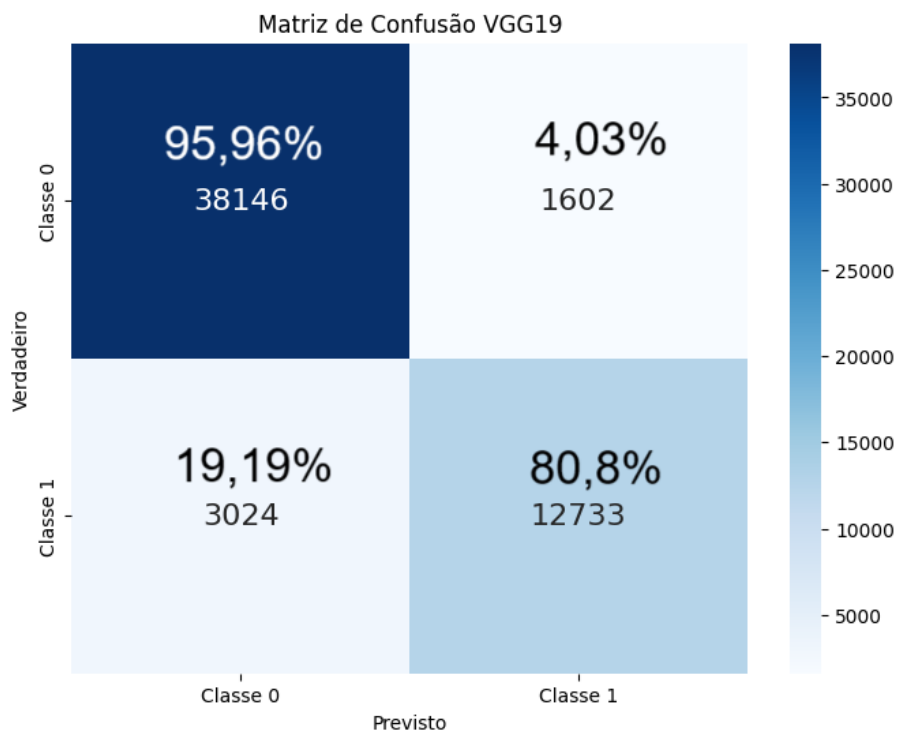


Figura 8. Matriz de Confusão VGG19.

Em relação a qual VGG teve o melhor desempenho, levando em consideração

a execução superior dos modelos de *CNN*, a matriz de confusão mostra que a VGG16, com uma taxa de 86,88% de acerto, foi melhor em classificar corretamente imagens com *IDC*. Por outro lado, a VGG19 foi melhor para classificação de imagens sem *IDC*, com uma taxa de 95,96% de acerto. Isto ocorre pelo fato de que a VGG16 é uma rede menos profunda em relação a quantidade de camadas, conseguindo ter um desempenho melhor em generalizar conjuntos de dados com menos exemplos, como por exemplo, as imagens com *IDC*, que representam apenas 28,38% da base de dados. Já a VGG19, por ser mais profunda, se ajusta melhor aos dados de treinamento, que em sua maioria são imagens sem *IDC*, obtendo desempenho inferior a classes menos prevalentes na base de dados [Mhaskar et al. 2017].

A fim de auxiliar e promover o diagnóstico médico, é de maior interesse utilizar a ferramenta com maior taxa de acertos em imagens com presença de *IDC*. Portanto, a VGG16 apresenta maior potencial para a classificação de tais imagens, representando 86% das predições corretas do diagnóstico positivo.

6. Conclusões e Trabalhos Futuros

Este trabalho focou na predição da presença de *IDC* em imagens de histologia de câncer de mama utilizando algoritmos de AP. Os principais objetivos do trabalho foram a implementação dos modelos de *CNN*, a verificação do uso desta ferramenta para o diagnóstico da patologia, a análise do desempenho do modelo por meio de métricas de avaliação e a comparação de seus resultados com algoritmos tradicionais de AM.

Os resultados mostraram que os modelos de *CNN* superaram os modelos tradicionais como FA, *XGBoost* e *KNN* em termos de acurácia, precisão, *recall* e *F1-score*. A melhor configuração para ambos os modelos de *CNN* utilizou o otimizador Adam com uma taxa de aprendizado de 0,0001, resultando em acurácia superior a 91% na base de testes. Os experimentos sugerem o potencial das *CNNs* para a tarefa de classificação de imagens com *IDC*, em especial a VGG16.

As contribuições incluem a avaliação da eficácia de *CNNs* na análise de imagens histológicas, sugerindo a capacidade dessas redes em lidar com variações sutis nas imagens e proporcionar suporte no processo de diagnóstico, ajudando a reduzir a variabilidade na interpretação das imagens. Isso pode resultar em diagnósticos mais rápidos e precisos, melhorando o prognóstico e a eficácia do tratamento para pacientes com câncer de mama.

Para trabalhos futuros, pretende-se preencher as lacunas que esta pesquisa deixou, como investigar técnicas de otimização e pré-processamento que possam melhorar o desempenho dos algoritmos tradicionais de AM. O uso de outras bases de dados a fim de testar a generalização dos modelos desenvolvidos. Por fim, a implementação de outras arquiteturas de AP para tarefa de classificação de imagens.

Referências

- [Agarap 2018] Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.
- [Aggarwal 2018] Aggarwal, C. C. (2018). *Neural Networks and Deep Learning: A Text-book*. Springer Publishing Company, Incorporated, 1st edition.

- [Basha et al. 2020] Basha, S. S., Dubey, S. R., Pulabaigari, V., and Mukherjee, S. (2020). Impact of fully connected layers on performance of convolutional neural networks for image classification. *Neurocomputing*, 378:112–119.
- [Chapala and Sujatha 2020] Chapala, H. and Sujatha, B. (2020). Resnet: Detection of invasive ductal carcinoma in breast histopathology images using deep learning. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 60–67.
- [Das et al. 2021] Das, A., Narayan Mohanty, M., Kumar Mallick, P., Tiwari, P., Muhammad, K., and Zhu, H. (2021). Breast cancer detection using an ensemble deep learning method. *Biomedical Signal Processing and Control*, 70:103009.
- [Deng et al. 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.
- [Ezzat and Hassanien 2023] Ezzat, D. and Hassanien, A. E. (2023). Optimized bayesian convolutional neural networks for invasive breast cancer diagnosis system. *Applied Soft Computing*, 147:110810.
- [Garzon and Barbosa 2023] Garzon, N. A. and Barbosa, L. S. d. O. (2023). Aprendizado de máquina na medicina: como algoritmos de aprendizado de máquina podem ser aplicados em diagnósticos médicos, prognósticos e descoberta de novos tratamentos. *RECIMA21 - Revista Científica Multidisciplinar - ISSN 2675-6218*, 4(12):e4124708.
- [Gennaro et al. 2022] Gennaro, L., Ribas, C. A. P. M., Ramos, R. K., Ariede, B. L., Waaga-Gasser, A. M., Collaço, L. M., Sigwalt, M. F., Ribas Filho, J. M., and Kubrusly, L. F. (2022). Invasive ductal breast carcinoma: Correlation of axl and -catenin immunohistochemical expression with tumor aggressivity. *SciELO Preprints*.
- [Goodfellow et al. 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- [Janowczyk and Madabhushi 2016] Janowczyk, A. and Madabhushi, A. (2016). Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. *Journal of Pathology Informatics*, 7(1):29.
- [Karatayev et al. 2021] Karatayev, M., Khalyk, S., Adai, S., Lee, M.-H., and Demirci, M. F. (2021). Breast cancer histopathology image classification using cnn. In *2021 16th International Conference on Electronics Computer and Computation (ICECCO)*, pages 1–5.
- [LeCun et al. 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521:436–44.
- [Ludermir 2021] Ludermir, T. B. (2021). Inteligência artificial e aprendizado de máquina: estado atual e tendências. *Estudos Avançados*, 35:85–94.
- [Mhaskar et al. 2017] Mhaskar, H., Liao, Q., and Poggio, T. (2017). When and why are deep networks better than shallow ones? *Proceedings of the AAAI Conference on Artificial Intelligence*, 31.

- [Mooney 2024] Mooney, P. (2024). Breast histopathology images. Disponível em <https://www.kaggle.com/datasets/paultimothymooney/breast-histopathology-images/data>. Acesso em: 2024-30-06.
- [Moreira 2023] Moreira, C. I. d. S. (2023). Criação de bases de dados de imagens histológicas anotadas e desenvolvimento de um modelo de classificação automática de patologias mamárias.
- [Sajid et al. 2023] Sajid, U., Khan, R. A., Shah, S. M., and Arif, S. (2023). Breast cancer classification using deep learned features boosted with handcrafted features. *Biomedical Signal Processing and Control*, 86:105353.
- [Santos et al. 2023] Santos, M. d. O., Lima, F. C. d. S. d., Martins, L. F. L., Oliveira, J. F. P., Almeida, L. M. d., and Cancela, M. d. C. (2023). Estimativa de incidência de câncer no brasil, 2023-2025. *Revista Brasileira de Cancerologia*, 69(1):e-213700.
- [Simonyan and Zisserman 2015] Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.
- [Smith 2017] Smith, L. N. (2017). Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472.
- [Srivastava et al. 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.
- [Yacoubby and Axman 2020] Yacoubby, R. and Axman, D. (2020). Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In Eger, S., Gao, Y., Peyrard, M., Zhao, W., and Hovy, E., editors, *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 79–91, Online. Association for Computational Linguistics.
- [Zhang et al. 2023] Zhang, A., Lipton, Z. C., Li, M., and Smola, A. J. (2023). *Dive into Deep Learning*. Cambridge University Press. <https://D2L.ai>.
- [Zhao et al. 2022] Zhao, H., Ming, X., and Yang, Z. (2022). Dcis (ductal carcinoma in situ)-like invasive ductal carcinoma of the breast. *Asian Journal of Surgery*, 45(10):1855–1856.