



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA
MESTRADO ACADÊMICO EM ENGENHARIA DE TELEINFORMÁTICA

ELENE FIRMEZA OHATA

**UMA ABORDAGEM PARA A CLASSIFICAÇÃO DE IMAGENS HISTOLÓGICAS
PARA O AUXÍLIO DA DETECÇÃO DO CÂNCER COLORRETAL BASEADO EM
*TRANSFER LEARNING***

FORTALEZA

2021

ELENE FIRMEZA OHATA

UMA ABORDAGEM PARA A CLASSIFICAÇÃO DE IMAGENS HISTOLÓGICAS PARA O
AUXÍLIO DA DETECÇÃO DO CÂNCER COLORRETAL BASEADO EM *TRANSFER*
LEARNING

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia de Teleinformática do Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas.

Orientador: Prof. Dr. Pedro Pedrosa Rebouças Filho

FORTALEZA

2021

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

O32a Ohata, Elene Firmeza.

Uma Abordagem para a Classificação de Imagens Histológicas para o Auxílio da Detecção do Câncer Colorretal baseado em Transfer Learning / Elene Firmeza Ohata. – 2021.
73 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2021.

Orientação: Prof. Dr. Pedro Pedrosa Rebouças Filho.

1. Transfer Learning. 2. Câncer Colorretal. 3. Imagens Histológicas. 4. Redes Neurais Convolucionais. I. Título.

CDD 621.38

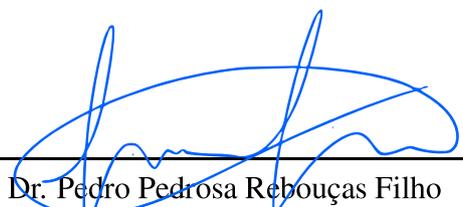
ELENE FIRMEZA OHATA

UMA ABORDAGEM PARA A CLASSIFICAÇÃO DE IMAGENS HISTOLÓGICAS PARA O
AUXÍLIO DA DETECÇÃO DO CÂNCER COLORRETAL BASEADO EM *TRANSFER*
LEARNING

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia de Teleinformática do Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas.

Aprovada em: 19 de agosto de 2021

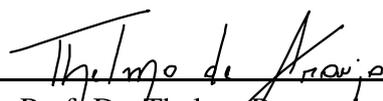
BANCA EXAMINADORA



Prof. Dr. Pedro Pedrosa Rebouças Filho (Orientador)
Universidade Federal do Ceará (PPGETI-UFC)



Prof. Dr. Ajalmar Rêgo da Rocha Neto
Universidade Federal do Ceará (PPGETI-UFC)



Prof. Dr. Thelmo Pontes de Araújo
Universidade Estadual do Ceará (UECE)

AGRADECIMENTOS

À minha mãe, Maria Elena, por me ensinar a sempre correr atrás dos meus objetivos.

Ao meu companheiro, Matheus, por seu amor, companheirismo, compreensão e incentivo durante essa etapa tão importante de minha vida.

Ao meu orientador, Prof. Dr. Pedro Pedrosa Rebouças Filho, por sua orientação, amizade, e ensinamentos durante o desenvolvimento deste trabalho.

Aos professores e a todos os meus amigos e companheiros de pesquisa do Programa de Pós-Graduação em Engenharia de Teleinformática da Universidade Federal do Ceará (PPGE-TI/UFC) e, em especial, do Laboratório de Processamento de Imagens, Sinais e Computação Aplicada (LAPISCO), que contribuíram para o meu crescimento profissional e pessoal.

À Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (FUN-CAP), pelo financiamento da pesquisa de mestrado.

A todos não mencionados acima que, de alguma forma, contribuíram para a realização deste trabalho.

RESUMO

O câncer colorretal (CRC) é o segundo tipo de câncer mais comum nos Estados Unidos. Este tipo de câncer é identificado por meio de avaliações histológicas de imagens microscópicas da região cancerosa, contando com uma interpretação subjetiva. O conjunto de dados de *Colorectal Cancer Histology* utilizado contém 5000 imagens e foi disponibilizado pela *University Medical Center Mannheim*. Esta abordagem propõe a identificação automática de oito tipos de tecidos encontrados na avaliação histológica do CRC. Foi aplicada a técnica chamada de *Transfer Learning* utilizando arquiteturas de Redes Neurais Convolucionais (CNNs). As estruturas das CNNs foram modificadas para extrair características das imagens e inserí-las em métodos de aprendizado de máquina, tais como: Naive Bayes, Perceptron de Múltiplas Camadas, k-Vizinhos mais Próximos, Floresta Aleatória e Máquina de Vetores de Suporte (SVM). Foram avaliadas 108 combinações extrator-classificador. A DenseNet169 combinada com o classificador SVM foi a que obteve o resultado mais satisfatório, atingindo uma Acurácia de 92,083% e F1-Score de 92,117%. Portanto, esta abordagem é capaz de distinguir tecidos encontrados na avaliação histopatológica do CRC.

Palavras-chave: *Transfer Learning*. Câncer Colorretal. Imagens Histológicas. Redes Neurais Convolucionais.

ABSTRACT

Colorectal cancer (CRC) is the second most diagnosed cancer in the United States. It is identified by histological evaluations of microscopic images of the cancerous region, relying on a subjective interpretation. The Colorectal Cancer Histology dataset used in this study contains 5000 images, made available by the University Medical Center Mannheim. This approach proposes the automatic identification of eight types of tissues found in CRC histopathological evaluation. We apply Transfer Learning from architectures of Convolutional Neural Networks (CNNs). We modified the structures of CNNs to extract features from the images and input them to well-known machine learning methods: Naive Bayes, Multilayer Perceptron, k-Nearest Neighbors, Random Forest, and Support Vector Machine (SVM). We evaluated 108 extractor-classifier combinations. The one that achieved the best results is DenseNet169 with SVM, reaching an Accuracy of 92.083% and F1-Score of 92.117%. Therefore, our approach is capable of distinguishing tissues found in CRC histological evaluation.

Keywords: Transfer Learning. Colorectal cancer. Histological Images. Convolutional Neural Networks.

LISTA DE FIGURAS

Figura 1 – Operação de convolução em uma imagem.	31
Figura 2 – Demonstração do processo de <i>max pooling</i> e <i>average pooling</i>	31
Figura 3 – Em (a), a ilustração representa o fluxo de informações convencionais durante uma passagem direta de uma rede e, em (b), esse fluxo é interrompido (definido como zero) em algum neurônio, consistindo no processo chamado de <i>Dropout</i>	32
Figura 4 – Metodologia da abordagem proposta.	44
Figura 5 – Amostras de cada classe do conjunto de dados <i>Colorectal Cancer Histology</i> . Da esquerda para a direita, conjuntos de oito imagens aleatórias para as classes (A) epitélio tumoral, (B) estroma simples, (C) estroma complexo, (D) células imunes, (E) detritos e muco, (F) glândula mucosa, (G) tecido adiposo, e (H) fundo.	45
Figura 6 – A imagem de entrada é submetida a uma sequência de camadas convolucionais e sub-amostragem (<i>pooling</i>). Os resultados são redimensionados e concatenados em uma única lista, chamada de <i>deep features</i>	48
Figura 7 – Resultado do teste de Nemenyi para os classificadores utilizados (Bayes, k-Vizinhos mais Próximos (<i>k-Nearest Neighbors</i>) (k-NN) - <i>Nearest_Neighbors</i> , Floresta Aleatória - <i>Random_Forest</i> , Perceptron Multicamadas (<i>Multilayer Perceptron</i>) (MLP), Máquina de Vetores de Suporte (<i>Support Vector Machines</i>) (SVM) (RBF) e SVM (Linear).	55
Figura 8 – Acurácia, F1-Score e tempo de teste para as melhores combinações de extrator de atributos com classificador. C-201: DenseNet201 + SVM (RBF), C-169: DenseNet169 + SVM (RBF).	59
Figura 9 – Matriz de confusão de DenseNet169 combinada com SVM RBF. <i>True label</i> - Classe verdadeira, <i>Predicted label</i> - Classe Predita. Classe 0 - epitélio tumoral, Classe 1 - estroma simples, Classe 2 - estroma complexo, Classe 3 - células imunes, Classe 4 - detritos e muco, Classe 5- glândula mucosa, Classe 6 - tecido adiposo e Classe 7 - fundo.	59

Figura 10 – Matriz de confusão da DenseNet201 combinada com SVM RBF. *True label* - Classe verdadeira, *Predicted label* - Classe Predita. Classe 0 - epitélio tumoral, Classe 1 - estroma simples, Classe 2 - estroma complexo, Classe 3 - células imunes, Classe 4 - detritos e muco, Classe 5- glândula mucosa, Classe 6 - tecido adiposo e Classe 7 - fundo. 62

Figura 11 – Visualização dos atributos extraídos com a DenseNet169 utilizando *t-Distributed Stochastic Neighbor Embedding* (t-SNE). Classe 0 - epitélio tumoral, Classe 1 - estroma simples, Classe 2 - estroma complexo, Classe 3 - células imunes, Classe 4 - detritos e muco, Classe 5- glândula mucosa, Classe 6 - tecido adiposo e Classe 7 - fundo. 63

LISTA DE TABELAS

Tabela 1 – Arquiteturas de Redes Neurais Convolucionais (<i>Convolutional Neural Networks</i>) (CNN) usadas como extratores de características e seus respectivos destaques.	42
Tabela 2 – Arquiteturas de CNN, as configurações utilizadas, o tamanho de imagem de entrada, e o número de atributos extraídos.	49
Tabela 3 – Configurações utilizadas na “busca” pelos hiperparâmetros dos classificadores.	50
Tabela 4 – Matriz de confusão para a classificação de imagens de câncer colorretal (CRC).	51
Tabela 5 – Acurácia e <i>F1-Score</i> obtidos por cada combinação de extrator e classificador de características. Seus respectivos desvios padrão acompanham todas as métricas.	56
Tabela 5 – (Cont.) Acurácia e <i>F1-Score</i> obtidos por cada combinação de extrator e classificador de características. Seus respectivos desvios padrão acompanham todas as métricas.	57
Tabela 5 – (Cont.) Acurácia e <i>F1-Score</i> obtidos por cada combinação de extrator e classificador de características. Seus respectivos desvios padrão acompanham todas as métricas.	58
Tabela 6 – Tempos de treino e teste obtidos por cada combinação de extrator e classificador de características. Seus respectivos desvios padrão acompanham todas as métricas.	60
Tabela 6 – (Cont.) Tempos de treino e teste obtidos por cada combinação de extrator e classificador de características. Seus respectivos desvios padrão acompanham todas as métricas.	61
Tabela 6 – (Cont.) Tempos de treino e teste obtidos por cada combinação de extrator e classificador de características. Seus respectivos desvios padrão acompanham todas as métricas.	62
Tabela 7 – Comparação dos resultados obtidos pelo método proposto e outras abordagens.	64

LISTA DE ABREVIATURAS E SIGLAS

k-NN	k-Vizinhos mais Próximos (<i>k-Nearest Neighbors</i>)
MLP	Perceptron Multicamadas (<i>Multilayer Perceptron</i>)
SVM	Máquina de Vetores de Suporte (<i>Support Vector Machines</i>)
t-SNE	<i>t-Distributed Stochastic Neighbor Embedding</i>
CNN	Redes Neurais Convolucionais (<i>Convolutional Neural Networks</i>)
CRC	câncer colorretal
IDH	Índice de Desenvolvimento Humano
H&E	Hematoxilina e Eosina
MAP	Máxima Probabilidade a Posteriori
ILSVRC	<i>ImageNet Large Scale Visual Recognition Challenge</i>
NAS	<i>Neural Architecture Search</i>
IoT	Internet das Coisas (<i>Internet of Things</i>)
GPU	Unidade de Processamento Gráfico (<i>Graphical Processing Unit</i>)
TP	verdadeiro positivo (<i>True Positive</i>)
FP	falso positivo (<i>False Positive</i>)
FN	falso negativo (<i>False Negative</i>)
TN	verdadeiro negativo (<i>True Negative</i>)
TE	Taxa de Erro

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Estado da arte	15
1.2	Objetivos	17
1.2.1	<i>Objetivos específicos</i>	17
1.3	Produção científica	17
1.4	Organização da dissertação	19
2	MÉTODOS DE APRENDIZAGEM DE MÁQUINA	20
2.1	Naive Bayes	20
2.2	k-Vizinhos mais Próximos	21
2.3	Floresta Aleatória	22
2.4	Máquina de Vetores de Suporte	24
2.4.1	<i>Classificador de Hiperplano</i>	24
2.4.2	<i>Classificador de Margem Máxima</i>	25
2.4.3	<i>Classificador de Margem Suave</i>	25
2.4.4	<i>Classificador SVM</i>	26
2.5	Perceptron de Múltiplas Camadas	27
3	REDES NEURAIAS CONVOLUCIONAIS (CNN)	29
3.1	Estrutura de uma Rede Neural Convolutacional	29
3.1.1	<i>Camada Convolutacional</i>	30
3.1.2	<i>Camada de Pooling</i>	30
3.1.3	<i>Dropout</i>	32
3.1.4	<i>Camada Totalmente Conectada</i>	32
3.1.5	<i>Softmax</i>	33
3.2	Arquiteturas VGG	33
3.2.1	<i>Convoluções Fatoradas</i>	33
3.3	Arquiteturas Inception	34
3.3.1	<i>Módulo Inception</i>	35
3.3.2	<i>Camada Batch-Normalization</i>	35
3.3.3	<i>Convoluções Assimétricas</i>	36
3.4	Arquiteturas ResNet	36

3.4.1	<i>Bloco Residual</i>	36
3.5	Arquiteturas Inception-ResNet	37
3.5.1	<i>Módulo Inception com Conexões Residuais</i>	37
3.6	Arquitetura Xception	37
3.6.1	<i>Convoluções Separáveis por Profundidade</i>	38
3.7	Arquitetura MobileNet	38
3.7.1	<i>Multiplicadores de Largura</i>	38
3.7.2	<i>Multiplicadores de Resolução</i>	39
3.8	Arquiteturas DenseNet	39
3.8.1	<i>Blocos Densos</i>	39
3.9	Arquiteturas NASNet	40
3.9.1	<i>NASNet Search Space</i>	40
3.10	<i>Transfer Learning</i>	40
4	METODOLOGIA	43
4.1	Conjunto de dados	43
4.2	Extração de características	45
4.2.1	<i>CNN como extrator de características</i>	46
4.2.1.1	<i>Construção e inicialização das arquiteturas</i>	46
4.2.1.2	<i>Treinamento das Arquiteturas</i>	47
4.2.1.3	<i>Pré-processamento</i>	47
4.2.1.4	<i>Convertendo CNN em extratores de características</i>	47
4.3	Classificação	49
4.4	Métricas de Avaliação	50
4.5	Validação cruzada	52
5	RESULTADOS E DISCUSSÕES	54
5.1	Análise de Desempenho	54
5.2	Comparação com outros métodos no conjunto de dados <i>Colorectal Cancer Histology</i>	63
6	CONCLUSÃO E TRABALHOS FUTUROS	65
	REFERÊNCIAS	66
	ANEXOS	72

1 INTRODUÇÃO

Entre 2008 e 2017, as taxas de incidência do Câncer Colorectal (CRC) aumentaram; as taxas de crescimento atingiram 2% ao ano, conseqüentemente, as taxas de mortalidade aumentaram 13% em adultos abaixo de 50 anos (SIEGEL *et al.*, 2017b). O CRC corresponde ao terceiro tipo de câncer mais diagnosticado e à quarta maior causa de mortes por câncer no mundo (FERLAY *et al.*, 2015). Enquanto que nos Estados Unidos da América, o CRC é segundo tipo de câncer mais detectado (SIEGEL *et al.*, 2020). Este tipo de câncer é comumente especificado de acordo com sua localização inicial, como câncer de cólon ou câncer retal, onde a incidência e mortalidade são de 30% a 40% mais altas em homens do que em mulheres (ABIR *et al.*, 2006; SIEGEL *et al.*, 2017a). Globalmente, estima-se que até 2030 a incidência do CRC aumentará 60%, atingindo valores acima de 2,2 milhões (ARNOLD *et al.*, 2017; SIEGEL *et al.*, 2020).

A distribuição de incidência e mortalidade do CRC é amplamente variada. Aproximadamente dois terços de todos os casos e 60% de todas as mortes ocorreram em países com alto Índice de Desenvolvimento Humano (IDH). Porém, o rápido aumento ocorre em países com valores de IDH médio-alto. Esta diferença acontece, principalmente, devido à concentração da população mundial nestes países, e a incentivos a detecção e prevenção precoce nos países desenvolvidos. Essa diversidade socioeconômica dos países promove variações de até 10 vezes nas taxas de incidência e mortalidade do CRC (ARNOLD *et al.*, 2017). Então, é necessário ressaltar a importância da busca por métodos eficientes e acessíveis para o diagnóstico deste câncer que atinge o mundo.

O diagnóstico do CRC é baseado em imagens microscópicas da área cancerosa. Essas imagens compreendem exames histológicos definidos por procedimentos que permitem qualidade na avaliação microscópica do tecido. No procedimento, o tecido é alocado com formalina neutra a 10%, umedecido em parafina e dividido manualmente por um micrótomo para produzir cortes com espessura de 4-5 μm (SLAOU; FIETTE, 2010). Os especialistas realizam a delimitação e classificação das regiões encontradas visualmente e, conseqüentemente, subjetivamente. A subjetividade no diagnóstico se deve à semelhança de alguns tipos de tecidos encontrados nos exames (ARAÚJO *et al.*, 2017). Devido a essa alta similaridade, a classificação visual dos tecidos pode ser comprometida pelo cansaço, estresse, experiência e capacidade humana, tornando-se necessários sistemas automáticos de classificação. Essa ferramenta de classificação pode ser aplicada em laboratórios de imagens médicas para cooperar com o trabalho de técnicos e médicos. Por exemplo, se a classificação do especialista for diferente da do sistema

automático, a imagem investigada será enviada para uma etapa de revisão, onde um especialista a investigará com maior atenção.

A avaliação histopatológica do CRC é comumente conhecida como avaliação histopatológica de Hematoxilina e Eosina (H&E), devido à aplicação de H&E em cortes desparafinados (ARAÚJO *et al.*, 2017; SLAOUI; FIETTE, 2010). Esta avaliação de seções de tecido coradas é complexa, uma vez que a estrutura examinada possui tecidos diferentes, incluindo tecidos tumorais. Com base nisso, o conjunto de dados *Colorectal Cancer Histology* (Histologia do Câncer Colorretal) foi proposto por Kather *et al.* (2016). Este conjunto de dados representa uma coleção de imagens microscópicas contendo os diferentes tipos de tecidos encontrados, no qual o desafio consiste em classificar os oito tipos de tecidos apresentados (KATHER *et al.*, 2016).

Devido à necessidade global de diagnóstico do CRC, técnicas que utilizam aprendizado de máquina profundo (*Deep Learning*) e *Transfer Learning* podem ser aplicadas para contribuir para um método eficaz de diagnóstico desse tipo de câncer. Algumas outras abordagens propuseram métodos utilizando estas técnicas para auxílio a diagnósticos, como o auxílio na identificação e classificação de traumas encontrados em acidentes vasculares cerebrais de tomografia computadorizada (JR *et al.*, 2019; SARMENTO *et al.*, 2019; SARMENTO *et al.*, 2020), a classificação de malignidade dos nódulos pulmonares (HUSSEIN *et al.*, 2017; NÓBREGA *et al.*, 2018), o diagnóstico de pacientes com doenças retinianas de cegueira tratáveis (CHOI *et al.*, 2017; KERMANY *et al.*, 2018), e o monitoramento contínuo, baseado em exames e no diagnóstico de doenças cardiovasculares (LOH; THEN, 2017; SANTOS *et al.*, 2020).

Na literatura, várias abordagens já propuseram alternativas para automatizar a classificação de exames histopatológicos, por exemplo, na classificação de núcleos celulares encontrados em imagens histopatológicas de câncer de mama (ALHINDI *et al.*, 2018; GECER *et al.*, 2018; WANG *et al.*, 2016; XU *et al.*, 2015), CRC (BAYRAMOGLU; HEIKKILÄ, 2016; KATHER *et al.*, 2016; KATHER *et al.*, 2019; SHABAN *et al.*, 2020; XU *et al.*, 2017), e câncer de pulmão (COUDRAY *et al.*, 2018; GIRARD *et al.*, 2016; YU *et al.*, 2016; YU *et al.*, 2020). Em resumo, essas abordagens podem ser divididas em dois tipos, abordagens baseadas em aprendizado de máquina supervisionado com métodos de extração de características tradicionais, e abordagens baseadas em CNN para a execução de ambas as etapas (extração de características e classificação).

A partir disso, desenvolvemos um método para classificação multi-classe utilizando extração profunda de atributos, *Transfer Learning* e classificação extensiva dos atributos usando

o conjunto de dados *Colorectal Cancer Histology*. Este trabalho propõe combinar CNN aliada ao conceito de *Transfer Learning*, o que permite que esse método seja aplicado como um extrator de características. Além disso, técnicas de aprendizado de máquina foram aplicadas para a classificação dos oito tipos de tecidos. Os métodos de aprendizado de máquina empregados foram: Naive Bayes, MLP, k-NN, Floresta Aleatória (*Random Forest*) e SVM.

Para ressaltar a eficácia da abordagem, uma avaliação de desempenho foi apresentada contendo diversas combinações entre várias arquiteturas CNN com os métodos de classificação considerados.

1.1 Estado da arte

Na literatura, várias abordagens têm proposto acelerar e automatizar o diagnóstico de exames histológicos. A análise dos tecidos encontrados nas imagens histológicas é amplamente difundida em diversas doenças, como câncer de mama, câncer colorretal e câncer de pulmão. Os vários usos da avaliação histopatológica (H&E) demonstram o vasto espaço para melhorias no uso de exames histológicos e a importância de contrastar essas abordagens.

No câncer de mama, Xu *et al.* (2015) propuseram uma detecção de núcleos baseada em *Stacked Sparse Autoencoder* (SSAE) instanciado por aprendizagem profunda, mas com uma quantidade desequilibrada de amostras nucleares e não nucleares. Wang *et al.* (2016) caracterizaram qualitativamente imagens a partir da segmentação dos núcleos, que posteriormente, com a extração das características de forma e textura, foram classificadas apenas em forma binária. Gecer *et al.* (2018) apresentaram a classificação em cinco categorias baseadas na concatenação de quatro CNN. Essas CNN forneceram uma mesclagem de mapas para rotular cada lâmina histopatológica apresentada e fazer a classificação proposta, resultando em um alto custo computacional.

Além disso, a abordagem de Coudray *et al.* (2018) realizou classificação dos subtipos mais prevalentes de câncer de pulmão, que são o adenocarcinoma (LUAD) e o carcinoma de células escamosas (LUSC), distinguindo-os dos tecidos saudáveis. Yu *et al.* (2020) também realizaram uma classificação de tecidos entre LUAD e LUSC, mais precisamente, a classificação de subtipos transcriptômicos em câncer de pulmão de células não pequenas (NSCLC). Outra abordagem de classificação é a proposta por Yu *et al.* (2016), onde foi baseada em métodos de aprendizado de máquina regularizados e extratores de características de imagens para determinar sobreviventes de curto ou longo prazo com LUAD e/ou LUSC.

Para CRC, Xu *et al.* (2017) propuseram a segmentação e multi-classificação com base no uso de *Transfer Learning* e no estudo da ativação na última camada oculta. Os autores aplicaram uma CNN treinada no banco de dados ImageNet como extrator de atributos e observaram individualmente a sensibilidade de cada neurônio que compõe a camada oculta.

Adicionalmente, Bayramoglu e Heikkilä (2016) propuseram a classificação dos núcleos das células de CRC em três classes: núcleos epiteliais, fibroblastos e núcleos inflamatórios. Esta abordagem utiliza aprendizagem profunda, *Transfer Learning* e ajuste fino (*fine tuning*) para auxiliar no diagnóstico de núcleos de células de CRC. As CNN utilizadas foram AlexNet, GoogLeNet, VGG-16 e GenderNet. Os autores usaram todas as camadas das CNN, exceto a última camada totalmente conectada, que foi adaptada para a classificação das três classes de núcleos celulares propostas.

A classificação dos tecidos encontrados na avaliação histopatológica do CRC é abordada por Kather *et al.* (2019) através do treinamento de uma CNN via *Transfer Learning*. A etapa de *Transfer Learning* utilizou um conjunto de dados de treinamento de 100 mil imagens histopatológicas, divididas em nove tipos de tecido. Outro trabalho de multi-classificação proposto por Kather *et al.* (2016) corresponde à análise de texturas de tecidos encontradas em imagens histopatológicas de CRC. Mais precisamente, este artigo é o responsável por apresentar o conjunto de dados utilizado no método apresentado neste trabalho. Na abordagem de Kather *et al.* (2016), a estratégia de extração e classificação de texturas foi baseada na escolha empírica de hiper-parâmetros, demonstrando uma fragilidade na seleção da melhor combinação alcançada.

Portanto, as abordagens acima podem ser divididas principalmente em dois tipos: classificação utilizando CNN e abordagens de aprendizado de máquina supervisionado com métodos de extração de características. Métodos baseados na morfologia celular, definidos por segmentação, classificação individual das regiões alvo e classificação final, também fazem parte das abordagens com CNN.

Os métodos estruturados em biomarcadores realizam a extração de características com foco em uma propriedade proeminente da imagem, como a textura, reduzindo assim o número total de características extraídas e classificadas. Em resumo, as etapas de extração mencionadas acima são devidamente dependentes da segmentação inicial da região e da escolha correta do parâmetro para a boa segmentação. A adição dessas etapas pode limitar o desempenho de um sistema para auxílio a diagnóstico médico.

1.2 Objetivos

O objetivo principal deste trabalho é o desenvolvimento de um sistema de classificação de imagens histológicas utilizando CNN baseadas no conceito de *Transfer Learning*.

1.2.1 Objetivos específicos

De modo mais específico, este trabalho tem os seguintes objetivos:

- Explorar várias arquiteturas de CNN como extratores de características (com base na noção de *Transfer Learning*);
- Extrair atributos, a partir do conjunto de imagem considerado, usando diversos modelos de CNN pré-treinadas no conjunto de dados ImageNet;
- Aplicar diferentes métodos de classificação supervisionada;
- Desenvolver um sistema rápido e eficiente para o auxílio ao diagnóstico médico para a classificação de imagens histológicas usando CNN, incorporando o conceito de *Transfer Learning*;
- Analisar a relevância da abordagem proposta utilizando métricas de avaliação, como Acurácia e F1-Score, além de parâmetros de tempo de processamento;
- Comparar os resultados obtidos com estudos já publicados para validação da abordagem proposta.

1.3 Produção científica

A seguir estão listadas as publicações diretamente relacionadas à pesquisa de mestrado descrita neste documento, sendo o primeiro artigo a produção gerada com o estudo desta dissertação e os demais sendo aplicações relacionadas:

1. OHATA, E. F.; CHAGAS, J. V. S. das; BEZERRA, G. M.; HASSAN, M. M.; ALBUQUERQUE, V. H. C. de; FILHO, P. P. R. A Novel Transfer Learning Approach for the Classification of Histological Images of Colorectal Cancer. **The Journal of Supercomputing**, Springer, p. 1–26, 2021. Qualis B2 (Engenharia IV).
2. OHATA, E. F.; BEZERRA, G. M.; CHAGAS, J. V. S. d.; NETO, A. V. L.; ALBUQUERQUE, A. B.; ALBUQUERQUE, V. H. C. d.; FILHO, P. P. R. Automatic Detection of Covid-19 Infection using Chest X-Ray Images Through Transfer Learning. **IEEE/CAA Journal of Automatica Sinica**, v. 8, n. 1, p. 239–248, 2021. JCR Impact Factor: 6.171.

3. SILVA, S. P. P. da; ALMEIDA, J. S.; OHATA, E. F.; RODRIGUES, J. J. P. C.; ALBUQUERQUE, V. H. C. de; FILHO, P. P. R. Monocular Vision Aided Depth Map From Rgb Images to Estimate of Localization and Support to Navigation of Mobile Robots. **IEEE Sensors Journal**, v. 20, n. 20, p.12040–12048, 2020. Qualis A1 (Engenharia IV).
4. FERNANDES, L. S.; SILVA, F. H.; OHATA, E. F.; MEDEIROS, A.; NETO, A. V. L.; NOGUEIRA, Y. L.; REGO, P. A.; FILHO, P. P. R. A Robust Automatic License Plate Recognition System for Embedded Devices. In: **SPRINGER Brazilian Conference on Intelligent Systems (BRACIS)**. [S. l.], 2020. p. 226–239.
5. MEDEIROS, A.; OHATA, E. F.; SILVA, F. H.; REGO, P. A.; FILHO, P. P. R. An Approach to Bi-Rads Uncertainty Levels Classification via Deep Learning with Transfer Learning Technique. In: **2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)**. [S. l.], 2020. p. 603–608.
6. SILVA, F. H. S.; MEDEIROS, A. G.; OHATA, E. F.; FILHO, P. P. R. Classification of Electroencephalogram Signals for Detecting Predisposition to Alcoholism using Computer Vision and Transfer Learning. In: **2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)**, 2020. p. 126–131.
7. BARROS, A. C. da S.; OHATA, E. F.; SILVA, S. P. P. da; ALMEIDA, J. S.; FILHO, P. P. R. An Innovative Approach of Textile Fabrics Identification from Mobile Images using Computer Vision based on Deep Transfer Learning. In: **2020 International Joint Conference on Neural Networks (IJCNN)**, 2020. p. 1–8.
8. FERREIRA, M. A. A.; SOUZA, L. F. de F.; SILVA, F. H. dos S.; OHATA, E. F.; ALMEIDA, J. S.; FILHO, P. P. R. Intelligent Industrial Iot System for Detection of Short-Circuit Failure in Windings of Wind Turbines. In: **2020 International Joint Conference on Neural Networks (IJCNN)**, 2020. p. 1–8.
9. ZHANG, H.; SILVA, F. H.; OHATA, E. F.; MEDEIROS, A. G.; FILHO, P. P. R. Bi-Dimensional Approach based on Transfer Learning for Alcoholism Predisposition Classification via Eeg Signals. **Frontiers in Human Neuroscience**, Frontiers, v. 14, p. 365, 2020. Qualis A1 (Interdisciplinar).
10. HU, Q.; OHATA, E. F.; SILVA, F. H.; RAMALHO, G. L.; HAN, T.; FILHO, P. P. R. A New Online Approach for Classification of Pumps Vibration Patterns based on Intelligent Iot System. **Measurement**, Elsevier, v. 151, p. 107138, 2020. Qualis A1 (Engenharia IV).
11. MEDEIROS, A. G.; SILVA, F. H.; OHATA, E. F.; PEIXOTO, S. A.; FILHO, P. P. R. An

Automatic Left Ventricle Segmentation on Echocardiogram Exams via Morphological Geodesic Contour with Adaptive External Energy. **Journal of Artificial Intelligence and Systems**, v. 1, p. 77–95, 2019.

12. ALMEIDA, J.; OHATA, E.; SILVA S.; SILVA F.; SOUSA, P.; NASCIMENTO, N.; SILVA, FILHO, P. P. R. A New Strategy for Mobile Robots Localization based on Omnidirectional Sonar Images and Machine Learning. In: **Anais Estendidos da XXXII Conference on Graphics, Patterns and Images (SIBGRAPI)**. Porto Alegre, RS, Brasil: SBC, 2019. p. 168–171.
13. OHATA, E.; SILVA, F. H.; ALVES, S.; SILVA, S.; ALMEIDA, J.; FILHO, P. P. R. An Approach to Classify Chronic Kidney Diseases using Scintigraphy Images. In: **Anais Estendidos da XXXII Conference on Graphics, Patterns and Images (SIBGRAPI)**. Porto Alegre, RS, Brasil: SBC, 2019. p.156–159.
14. ALVES, S. S. A. A.; OHATA, E. F.; NASCIMENTO, N. M. M.; SOUZA, J. W. M. d.; HOLANDA, G. B.; LOUREIRO, L. L.; FILHO, P. P. R. Gender-based approach to estimate the human body fat percentage using Machine Learning. In: **2021 International Joint Conference on Neural Networks (IJCNN)**, 2021. Apresentado e aguardando anais de evento.

1.4 Organização da dissertação

Além do capítulo de Introdução (Capítulo 1), esta dissertação apresenta outros cinco capítulos. O Capítulo 2 apresenta métodos de aprendizado de máquina consolidados na literatura, os quais foram usados para avaliar a abordagem proposta. O Capítulo 3 traz as principais arquiteturas de CNN e seus conceitos fundamentais. O Capítulo 4 descreve a metodologia da abordagem desenvolvida. O Capítulo 5 apresenta os resultados experimentais obtidos nesta pesquisa. Finalmente, o Capítulo 6 contém as conclusões do estudo e as propostas para trabalhos futuros.

2 MÉTODOS DE APRENDIZAGEM DE MÁQUINA

Neste capítulo, os classificadores empregados nesta dissertação são apresentados: Naive Bayes (Seção 2.1), k-NN (Seção 2.2), Floresta Aleatória (Seção 2.3), SVM (Seção 2.4) e MLP (Seção 2.5).

2.1 Naive Bayes

Em Aprendizagem de Máquina, há um conjunto de classificadores probabilísticos simples conhecidos como Naive Bayes. Com base no teorema de Bayes, esses classificadores assumem independência condicional de classe entre os atributos. Em outras palavras, um classificador Naive Bayes presume que, para uma dada classe, a probabilidade de presença (ou ausência) de uma determinada característica não está relacionada à presença de qualquer outra característica (THEODORIDIS *et al.*, 2010). De modo mais formal, o teorema de Bayes afirma a seguinte relação:

$$P(c|f_1, \dots, f_k) = \frac{P(c) \cdot P(f_1, \dots, f_k|c)}{P(f_1, \dots, f_k)}, \quad (2.1)$$

em que c é a variável de classe, $P(c)$ pode ser interpretada como a frequência relativa da classe no conjunto de treinamento (Probabilidade *a Priori*), k é o número de atributos e de f_1 a f_k são os elementos vetoriais de atributos dependentes. Com base na proposição da independência, pode-se afirmar que:

$$P(f_i|c, f_1, \dots, f_{i-1}, f_{i+1}, \dots, f_k) = P(f_i|c). \quad (2.2)$$

Assim, o teorema de Bayes pode ser reconfigurado como:

$$P(c|f_1, \dots, f_k) = \frac{P(c) \cdot \prod_{i=1}^k P(f_i|c)}{P(f_1, \dots, f_k)}. \quad (2.3)$$

No entanto, como $P(f_1, \dots, f_k)$ é constante para todas as classes possíveis, a Equação 2.3 pode ser ainda mais simplificada, utilizando a regra de decisão de Máxima Probabilidade a Posteriori (MAP), a qual é descrita matematicamente como:

$$P(c|f_1, \dots, f_k) \propto P(c) \cdot \prod_{i=1}^k P(f_i|c) \quad (2.4)$$

$$\hat{c} = \underset{c}{\operatorname{argmax}} P(c) \cdot \prod_{i=1}^k P(f_i|c) \quad (2.5)$$

Isso significa que, por exemplo, para classificar corretamente uma imagem, será necessário executar três etapas. Primeiro, para cada classe, será calculado o produto da probabilidade de ocorrência das características (probabilidade a posteriori). Em seguida, haverá a multiplicação dessas probabilidades pela frequência relativa (no conjunto de treinamento) de sua respectiva classe (probabilidade a priori). Por fim, será selecionada a classe com maior probabilidade.

Além disso, como mencionado anteriormente, existem vários classificadores Naive Bayes. O utilizado nesta dissertação foi o Naive Bayes Gaussiano, que basicamente assume que a probabilidade a priori, ou mesmo função densidade de probabilidade, tem distribuição Gaussiana, também conhecida como distribuição Normal, e pode ser matematicamente descrita como:

$$P(f_i|c) = \frac{e\left(-\frac{(f_i-\mu_c)^2}{2\sigma_c^2}\right)}{\sqrt{2\pi\sigma_c^2}} \quad (2.6)$$

em que μ_c e σ_c representam a média e o desvio padrão de uma determinada classe e são calculados a partir das amostras no conjunto de treinamento.

2.2 k-Vizinhos mais Próximos

k-NN é um método de aprendizado de máquina supervisionado, fundamentado em um algoritmo de aprendizado não-paramétrico e baseado em instâncias (FUKUNAGA; NARENDRA, 1975). Ao não fazer suposições explícitas sobre a função de densidade de probabilidade dos dados, o k-NN impede, ao mesmo tempo, o processo de aprender essa distribuição e os perigos potenciais de modificá-la. No entanto, vale a pena ressaltar que esse treinamento/aprendizado mínimo traz custos consideráveis em relação à demanda de memória e ao tempo de computação. A demanda de memória é um fator crítico, pois esse classificador precisa armazenar um número potencialmente alto de instâncias (amostras com seus atributos e rótulos). Além disso, como é necessário processar todo o conjunto de dados para prever a classe de uma determinada observação, o tempo de computação também torna-se um fator crítico para o algoritmo k-NN.

O processo de inferência do k-NN pode ser descrito, grosso modo, como um processo de votação majoritária entre as k amostras de treinamento mais próximas da instância invisível, em que o voto de uma determinada amostra de treinamento é o rótulo/classe e quanto mais perto significa que o valor em uma determinada métrica de dissimilaridade é menor, entre elas as distâncias euclidiana, Mahalanobis, Minkowski, Hamming e outras. A distância euclidiana, por exemplo, é uma das métricas de dissimilaridade mais populares e pode ser formulada como:

$$d(X, X^i) = \sqrt{(x_1 - x_1^i)^2 + \dots + (x_p - x_p^i)^2} \quad (2.7)$$

em que X e X^i são dois pontos/instâncias arbitrários em um espaço dimensional p e d .

Formalmente, dado que X é uma observação invisível e $k \in N^*$, o algoritmo de classificação k-NN pode ser dividido em três etapas. Na primeira etapa, o k-NN calcula a distância entre X e cada amostra de treinamento (as k amostras de treinamento com valores mais baixos de distância pertencem a um conjunto A). Na segunda etapa, para cada classe possível, a probabilidade condicional é estimada. De forma específica, a probabilidade de uma observação invisível X pertencer à classe c é dada pela taxa de amostras do subconjunto A (com k elementos) que possuem um rótulo igual a c . Por fim, a instância invisível é rotulada com a classe mais provável (valor mais alto).

O processo de estimativa descrito para a segunda etapa de classificação do k-NN pode ser matematicamente descrito como:

$$P(c) = \frac{1}{K} \sum_{i \in A} I(y_i, c) \quad (2.8)$$

em que c é uma determinada classe do problema, y_i é um determinado rótulo da amostra A , A é o conjunto das k amostras de treinamento mais próximas e I é uma função indicadora que retorna 1 (um), quando suas entradas são iguais, ou 0 (zero), caso contrário.

2.3 Floresta Aleatória

Proposta pela primeira vez por Ho (1995), Floresta Aleatória (*Random Florest*) é um método de conjunto que combina as estimativas realizadas por um conjunto de Árvores de Decisão (*Decision Trees*) (BREIMAN, 2001). Em geral, as Árvores de Decisão são um algoritmo de aprendizagem de máquina do tipo supervisionado não-paramétrico utilizado para

tarefas de classificação e regressão. No contexto da classificação, seu objetivo é estimar o rótulo de uma determinada instância usando regras de decisão (limites) aprendidas com os dados de treinamento.

O algoritmo de treinamento das Árvores de Decisão produz uma solução sub-ótima e pode ser associado conceitualmente a uma busca gananciosa, na qual são tomadas decisões locais em cada nó da árvore. Outra desvantagem das Árvores de Decisão é a suscetibilidade de aprender também o ruído no conjunto de treinamento (*overfitting*).

Para reduzir a variação das predições das Árvores de Decisão, Floresta Aleatória usa uma abordagem de conjunto chamada *Random Bagging*. Nesse contexto, o *Random Bagging* é construído e treina vários estimadores (por exemplo, Árvores de Decisão) em diferentes subconjuntos aleatórios do conjunto de treinamento e, em seguida, combina seus resultados (por exemplo, estima o voto da maioria) para compor a predição final. Mais formalmente, acontece que cada Árvore de Decisão pode ser vista como uma vizinhança ponderada (*weighted neighborhood*) (LIN; JEON, 2006) e é expressa como:

$$\hat{y} = \sum_{i=1}^n W(X_i, X') y_i, \quad (2.9)$$

onde X' é uma observação invisível, \hat{y} é seu rótulo estimado, X_i é a i ésima amostra de treinamento, y_i é seu respectivo rótulo e W é a função de peso.

Essa função de peso é alterada de acordo com o método de classificação usado, por exemplo, no contexto de Árvores de Decisão:

$$W(X_i, X'_i) = \begin{cases} \frac{1}{k}, & \text{se } X_i \in K \\ 0, & \text{caso contrário} \end{cases} \quad (2.10)$$

onde k são pontos de K na mesma folha que X' . Além disso, é possível estender a Equação 2.10 para compor a predição da Floresta Aleatória, que é dada por:

$$\hat{y} = \frac{1}{m} \sum_{j=1}^m \sum_{i=1}^n W_j(X_i, X'_i) y_i, \quad (2.11)$$

onde m é o número de estimadores (árvores) usados e W_j são suas respectivas funções de peso.

2.4 Máquina de Vetores de Suporte

Para entender o classificador SVM é importante compreender os conceitos de classificador de hiperplano, classificador de margem máxima e classificador de margem suave.

2.4.1 Classificador de Hiperplano

Primeiramente, suponha-se que cada instância de um determinado problema possa ser representada como um ponto X em um espaço dimensional p , onde cada dimensão/eixo desse espaço está associado a um recurso desse problema. Este ponto pode ser formulado como:

$$X = (x_1, x_2, \dots, x_p). \quad (2.12)$$

Em seguida, também suponha-se que exista um hiperplano $p - 1$ dimensional que separe perfeitamente um conjunto de instâncias de um determinado problema por seus rótulos. Sua definição matemática é dada pela equação:

$$\beta_0 + \beta_1 \alpha_1 + \dots + \beta_p \alpha_p = 0, \quad (2.13)$$

onde β_0, \dots, β_p são os coeficientes do hiperplano $\alpha_1, \dots, \alpha_p \in P_o$, e P_o é um ponto cruzado pelo hiperplano.

Portanto, de acordo com a posição relativa de um determinado ponto X para um determinado hiperplano, pode-se definir uma função de indicador que use esse hiperplano como limite. Esta regra de decisão é dada por:

$$y = \begin{cases} +1, & \text{se } \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p > 0 \\ -1, & \text{se } \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p < 0 \end{cases} \quad (2.14)$$

onde os pontos do lado positivo do hiperplano estão associados a um rótulo positivo ($y = +1$) e o oposto também é válido.

Em seguida, o Classificador de Hiperplano usa a Equação 2.14 como um processo de estimativa, onde os pontos são amostras invisíveis e sua posição relativa ao hiperplano define a qual classe eles pertencem.

2.4.2 *Classificador de Margem Máxima*

Não obstante, se um conjunto de amostras for linearmente separável (perfeitamente separado usando um hiperplano), um número infinito desses hiperplanos poderá existir. Para selecionar os melhores hiperplanos dentre esses, foi proposto um método chamado Classificador de Margem Máxima.

O principal intuito por trás do Classificador de Margem Máxima é que o hiperplano de margem máxima, que consiste no hiperplano mais distante de todas as observações de treinamento, é a decisão ideal. O termo “mais distante” representa a soma com o maior valor da distância perpendicular de todas as amostras de treinamento a um determinado hiperplano. Em contrapartida, a menor dessas distâncias perpendiculares é conhecida como largura da margem.

Portanto, encontrar esse melhor hiperplano de separação que pode ser formulado como um problema de maximização:

$$\underset{\beta}{\text{maximize}} M, \quad (2.15)$$

$$y_i(x_i^T \beta + \beta_0) \geq M, \forall i = 1, \dots, n. \quad (2.16)$$

Observa-se que M é o comprimento da margem, n é o número de amostras de treinamento, $X_1, \dots, X_n \in R^P$ e rótulos de classe $y_1, \dots, y_n \in \{1, +1\}$.

O hiperplano resultante dessa otimização é conhecido como hiperplano de margem máxima. Para criar o Classificador de Margem Máxima, esse hiperplano ideal deve ser aplicado à Equação 2.14.

2.4.3 *Classificador de Margem Suave*

No entanto, uma das principais desvantagens do Classificador de Margem Máxima é sua sensibilidade a observações individuais. Para superar isso, o Classificador de Margem Suave, também conhecido como Classificador de Vetor de Suporte, adiciona uma variável de folga ε para cada observação no conjunto de treinamento. O objetivo dessa variável é permitir que algumas amostras sejam classificadas incorretamente se isso aumentar a margem do classificador.

Portanto, o processo de otimização para encontrar o melhor Classificador de Margem Suave para separar a maior parte das amostras do conjunto de treinamento pode ser modelado como o seguinte problema de otimização:

$$\underset{\beta_1, \dots, \beta_p, \varepsilon_1, \dots, \varepsilon_n}{\text{maximize}} \quad M, \quad (2.17)$$

$$y_i(x_i' \beta + \beta_0) \geq M(1 - \varepsilon_i), \quad (2.18)$$

$$\varepsilon_i \geq 0, \sum_{i=1}^n \varepsilon_i \leq C, \quad (2.19)$$

onde C é um hiperparâmetro maior que zero.

2.4.4 Classificador SVM

Uma das principais limitações do Classificador de Margem Suave é que seu limite de decisão é sempre linear. Para estender seus conceitos a problemas não-linearmente separáveis, o SVM é proposto. A ideia principal do classificador SVM é transformar o espaço dos atributos de entrada através de uma maneira específica, para que um limite de decisão linear (nesse novo espaço) possa separar corretamente as amostras. Como descrever exatamente como o Classificador de Vetores de Suporte é calculado está além do escopo desta dissertação, verifica-se que a regra de decisão do SVM pode ser matematicamente descrita como:

$$f(X) = \beta_0 + \sum_{i=1}^n \alpha_i K(X, X'_i), \quad (2.20)$$

onde α é um coeficiente do SVM aprendido durante o treinamento e K é a transformação do espaço de destino. Por exemplo, um *kernel* linear (transformação) pode ser expresso por:

$$K(X, X') = \sum_{j=1}^p X_j^T X'_j, \quad (2.21)$$

e a transformação radial é definida como:

$$K(X, X') = e^{\left(-\gamma \sum_{j=1}^p (X_j^T X'_j)\right)^2}. \quad (2.22)$$

2.5 Perceptron de Múltiplas Camadas

O MLP é um método de aprendizagem de máquina do tipo supervisionado não-paramétrico. Além disso, esse classificador também é uma extensão do classificador Perceptron para conjuntos de dados não-linearmente separáveis (HAYKIN, 2001).

Introduzido por Rosenblatt (1958), o algoritmo Perceptron calcula uma soma ponderada das entradas e aplica o resultado a uma função de ativação. Abstratamente, cada peso indica a relevância de um determinado recurso para prever uma determinada classe. Mais formalmente, a previsão de Perceptron pode ser expressa como mostra a Equação 2.23.

$$\hat{y} = \varphi \left(\sum_{i=1}^p x_i w_i + b \right), \quad (2.23)$$

onde φ é uma função de ativação (por exemplo, logística/*softmax*), x_i é um vetor de entrada, p é o número de atributos de uma determinada entrada, w_i é o vetor de pesos, que combinado com b , são parâmetros aprendíveis do Perceptron. Além disso, o processo de treinamento desse classificador é realizado de forma iterativa, fornecido por:

$$w_i(t+1) = w_i(t) + \eta(y - \hat{y}(t))x_i, \forall i = 1, \dots, p, \quad (2.24)$$

em que t é uma determinada etapa do tempo de treinamento, y é o rótulo verdadeiro, \hat{y} é o rótulo previsto e η é a taxa de aprendizado. No entanto, apesar de ter sido um algoritmo simples e poderoso, o Perceptron não é capaz de executar uma classificação não linear.

Para superar isso, o MLP foi proposto. Esse classificador consiste em um gráfico computacional composto por uma pilha de camadas de perceptron. Cada uma dessas camadas, exceto a última, possui suas funções de ativação logística, as quais são empregadas para criar um classificador linear, substituídas por outra função não-linear, como a função sigmoide, que é usada para executar uma transformação não-linear no espaço de atributos. Mais formalmente, a regra de decisão do MLP pode ser matematicamente formulada como:

$$\hat{y}_{l_k} = \begin{cases} \varphi_{0_k} \left(\sum_{i=1}^p x_i w_{0_{k_i}} + b_{0_k} \right), & \text{se } l = 0 \\ \varphi_{l_k} \left(\sum_{i=1}^D \hat{y}_{(l-1)_i} w_{l_{k_i}} + b_{l_k} \right), & \text{caso contrário} \end{cases} \quad (2.25)$$

onde x é o vetor de entrada de p dimensões, l é um determinado índice de camada, D é o número de neurônios na camada $l - 1$ e φ é a função não-linear. Vale ressaltar que, em geral, os parâmetros aprendíveis do MLP (pesos e viés) são inicializados com valores aleatórios próximos de zero.

Além disso, para melhorar seu desempenho, MLP geralmente utiliza matematicamente um processo iterativo, expresso pela equação:

$$\hat{w}_{l_{k_i}}(t+1) = \begin{cases} w_{0_{k_i}}(t) + \eta \cdot \delta_{l_i}(t) \cdot x_k(t), & \text{se } l = 0 \\ w_{l_{k_i}}(t) + \eta \cdot \delta_{(l+1)_i}(t) \cdot \hat{y}_{l_k}(t), & \text{caso contrário} \end{cases}, \quad (2.26)$$

onde η é a taxa de aprendizado e $\delta_{l_k}(t)$ é o gradiente local do k -ésimo neurônio da camada l -ésima. Esse gradiente local é dado por:

$$\delta_{l_k} = \begin{cases} \varphi'_{l_k}[u_{l_k}(t)] \frac{1}{2M} \sum_{o=1}^M [y_{l_o} - \hat{y}_{l_o}(t)]^2, & \text{se } l = z \\ \varphi'_{l_k}[u_{l_k}(t)] \sum_{j=1}^J w_{l_{k_i}}(t) \cdot \delta_{(l+1)_j}(t), & \text{caso contrário} \end{cases}, \quad (2.27)$$

onde z é a última camada, M é o número de neurônios na última camada (z) e J é o número total de neurônios na camada l -ésima.

Observe que a derivada $\varphi'_i[u_{l_k}(t)]$ possui valores diferentes, dependendo da função de ativação escolhida, por exemplo:

$$\varphi'_i[u_{l_k}(t)] = \begin{cases} y_{l_k}(t)[1 - y_{l_k}(t)], & \text{se logística/sigmoide} \\ \frac{1 - (y_{l_k}(t))^2}{2}, & \text{se tangente hiperbólica} \end{cases} \quad (2.28)$$

3 REDES NEURAS CONVOLUCIONAIS (CNN)

Neste capítulo, é apresentada uma visão geral sobre os blocos de construção essenciais das modernas arquiteturas de CNN com seu princípio de funcionamento e modelos.

Uma CNN consiste em um algoritmo de Aprendizagem Profunda proposto por LeCun *et al.* (1989) que somente tornou-se popular quando Krizhevsky *et al.* (2012) obtiveram bons resultados no 2012 *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC). Esse desafio consiste em resolver uma tarefa de classificação com 1000 classes e mais de 1 milhão de amostras (KRIZHEVSKY *et al.*, 2009; DENG *et al.*, 2009). Desde então, várias arquiteturas de CNNs foram propostas na tentativa de superar os resultados obtidos por Krizhevsky *et al.* (2012), dentre elas: VGG (SIMONYAN; ZISSERMAN, 2014), MobileNet (HOWARD *et al.*, 2017), ResNet (HE *et al.*, 2016a), NasNet (ZOPH; LE, 2016), Xception (CHOLLET, 2017) e DenseNet (HUANG *et al.*, 2017).

Em uma CNN, uma convolução representa uma operação linear que implica na multiplicação de um conjunto de pesos com a entrada, bastante similar com uma rede neural tradicional (LECUN *et al.*, 1989). Considerando que o método foi desenvolvido para entrada bidimensional, a multiplicação é feita entre uma matriz de dados de entrada e uma matriz bidimensional de pesos, a qual é chamada de filtro ou *kernel*, do inglês (LECUN *et al.*, 1989).

CNN conseguem manter as correlações entre os *pixels* vizinhos da imagem, uma vez que opera sobre campos receptivos locais, efetuando o processo de convolução. Logo, ela consegue diminuir a sensibilidade à translação, à rotação e à distorção das imagens (LECUN *et al.*, 1998). Outros tipos de redes neurais não têm a capacidade de preservar esse tipo de relação, visto que consideram as imagens como uma entrada unidimensional.

3.1 Estrutura de uma Rede Neural Convolutiva

As CNN são redes neurais *feedforward*, na qual sua arquitetura foi elaborada para tornar a rede invariante à translação, à rotação e à distorção da imagem de entrada (LECUN *et al.*, 1998). Elas são formadas por conexões locais, por pesos compartilhados, pela camada de *pooling* e outras diferentes camadas (LECUN *et al.*, 2015). A organização da rede é dada por camadas com funções distintas. A camada convolutiva é responsável pela extração das características da imagem. A camada de *pooling* efetua uma sub-amostragem da imagem, enquanto as camadas totalmente conectadas têm como tarefa interpretar as características extraídas.

3.1.1 Camada Convolutacional

Na camada de convolução, as unidades são dispostas em *feature maps* (mapas de características), na qual cada unidade está ligada a uma porção da camada precedente através de um conjunto de pesos denominados filtros ou *kernel* (LECUN *et al.*, 2015). Os neurônios da rede empregam um método chamado pesos compartilhados, que equivale a diversas conexões sendo manipuladas por um mesmo parâmetro (peso) (LECUN *et al.*, 1989).

Esse método restringe consideravelmente a quantidade de parâmetros da rede. Além disso, esse modelo de organização das conexões configura o processo de convolução, visando extrair determinadas características das imagens, dentre elas as linhas e os contornos. Os filtros também podem consistir de duas ou três dimensões. A cada instante, eles ligam-se a uma certa parte de uma matriz, denominada de campo receptivo local. Nessa situação, os filtros podem ser interpretados como pequenos quadrados que deslizam sobre uma matriz, realizando a operação de convolução. Esses filtros têm um tamanho definido, como 5×5 ou $5 \times 5 \times 3$ (para entradas tridimensionais) e sua ação de deslocamento sobre a matriz de *pixels* da imagem é chamada de *stride*.

A Figura 1 apresenta a aplicação de um filtro (2×2) em uma imagem (3×3), resultando em um *feature map* (2×2), utilizando um *stride* de 1 (um). O cálculo do tamanho do *feature map* é realizado segunda a Equação 3.1:

$$T_m = \frac{M - k}{s} + 1 \quad (3.1)$$

onde T_m representa o tamanho do *feature map*, M é a dimensão da imagem, k é o tamanho do filtro e s consiste no passo do deslocamento (*stride*). As variáveis T_m , M e k correspondem ao tamanho em uma das dimensões (horizontal ou vertical), mas é padrão o uso de imagens e filtros quadrados (altura e largura iguais). Desse modo, as variáveis representam os tamanhos em ambas as dimensões.

3.1.2 Camada de Pooling

Como a camada convolutacional não reduz o tamanho da entrada, um passo extra pode ser necessário para diminuir a saída da convolução, preferencialmente sem adicionar novos parâmetros ao modelo. Por isso, é empregada a camada de *pooling*. Essa camada tem como

Figura 1 – Operação de convolução em uma imagem.

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

$$0 \times 0 + 1 \times 1 + 3 \times 2 + 4 \times 3 = 19$$

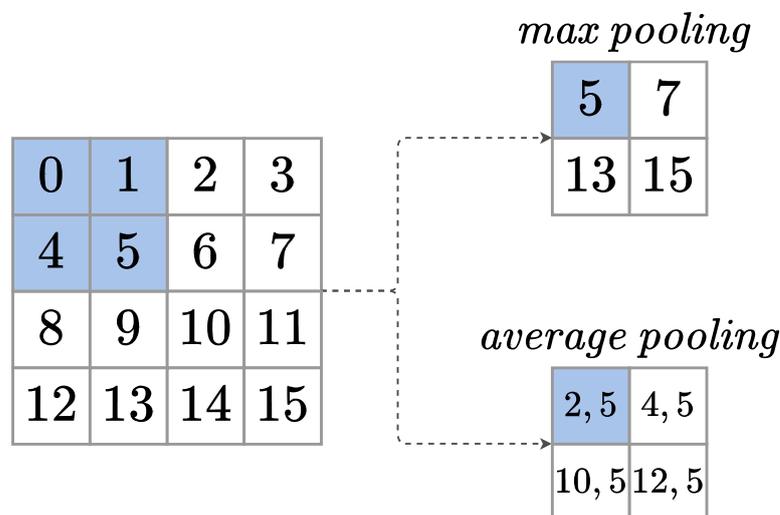
Fonte: Elaborado pelo autor.

objetivo alcançar um determinado nível de invariância ao deslocamento, diminuindo a resolução dos *feature maps* (GU *et al.*, 2018).

A camada de *pooling* opera de modo similar a uma camada convolutiva, contudo, o *stride* normalmente tem o mesmo tamanho do filtro (ao contrário da camada anterior, onde o *stride* era um). Dessa forma, os campos receptivos tornam-se totalmente distintos e a redução dos *features maps* alcança em torno de 75%. Os tipos mais comuns de *pooling* são o *max pooling* e o *average pooling*. No *max pooling*, o maior valor do campo receptivo é selecionado, enquanto no *average pooling*, a média dos valores é calculada (GOODFELLOW *et al.*, 2016)

Como pode ser visto na Figura 2, a camada de *pooling* tem como entrada uma imagem e efetua a operação de *max pooling* ou *average pooling* de 2×2 sobre a imagem. A saída retornada é uma imagem com a metade da altura e metade da largura original.

Figura 2 – Demonstração do processo de *max pooling* e *average pooling*.



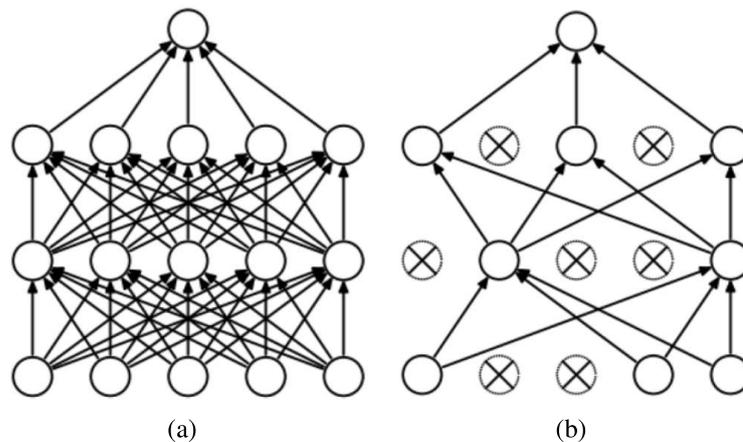
Fonte: Elaborado pelo autor.

3.1.3 Dropout

É bastante comum que uma rede aprenda a classificar muito bem as imagens de seu conjunto de treinamento, todavia, em outras imagens, ela pode ter uma queda no seu desempenho. Para evitar que ocorra esse sobreajuste (em inglês, *overfitting*) em seu aprendizado, utiliza-se a técnica de *Dropout*.

O método de *Dropout* consiste em eliminar aleatoriamente algumas unidades da rede neural (SRIVASTAVA *et al.*, 2014), como mostrado na Figura 3. De acordo com essa ideia, esses neurônios desativados (definidos como zero) não contribuem para o encaminhamento da rede e, conseqüentemente, não têm seus pesos atualizados durante a propagação reversa do gradiente. Uma probabilidade p é estabelecida e, assim, cada unidade tem essa probabilidade de seu sinal ser propagado na rede.

Figura 3 – Em (a), a ilustração representa o fluxo de informações convencionais durante uma passagem direta de uma rede e, em (b), esse fluxo é interrompido (definido como zero) em algum neurônio, consistindo no processo chamado de *Dropout*.



Fonte: (SRIVASTAVA *et al.*, 2014)

3.1.4 Camada Totalmente Conectada

A imagem de entrada é processada pelas camadas convolutivas e de *pooling*, que distinguem desde as características mais simples até às mais complexas, chegando nas camadas totalmente conectadas. Geralmente, a camada anterior é uma camada de *pooling*, na qual seus *feature maps* dispõem de mais de uma dimensão e, por isso, são redimensionados para apenas uma dimensão, a qual é representada em forma de vetor. Dessa forma, é possível conectar-

se à parte final da rede. As camadas totalmente conectadas operam como uma rede neural multicamadas *feedforward* e são responsáveis pela interpretação das características extraídas pelas camadas iniciais (LECUN *et al.*, 1989).

3.1.5 Softmax

A última camada é responsável por apresentar os resultados. O número de neurônios dispostos nessa camada é determinado pela quantidade de classes do problema. Cada neurônio exibe a saída em forma de probabilidade (0 a 1) e a resposta consiste no número do neurônio com saída de maior valor. Esse processo é realizado a partir da função *Softmax*, a qual é descrita pela Equação 3.2.:

$$y_j = \frac{e^{y(j)}}{\sum_{i=1}^n e^{y(i)}} \quad (3.2)$$

onde a saída y do neurônio j passa a ser o valor dessa mesma saída dividido pelo somatório de todas as saídas dos neurônios dessa camada.

3.2 Arquiteturas VGG

As experiências realizadas no artigo do *Oxford Visual Geometry Group* (VGG) (SIMONYAN; ZISSERMAN, 2014), incluindo o modelo vencedor das tarefas de detecção e localização do 2014 ILSVRC (RUSSAKOVSKY *et al.*, 2015), demonstraram que, para todas as arquiteturas avaliadas (11, 13, 16 ou 19 camadas), existe uma correlação positiva entre o aumento da profundidade (número de camadas) dos modelos e sua precisão no *ImageNet Dataset*. Esse aumento, sem superajustar o modelo, somente foi possível através de uma estratégia de regularização denominada Convoluções Fatoradas.

3.2.1 Convoluções Fatoradas

Inspirado no artigo Zeiler e Fergus (2014), que usava um tamanho de janela receptivo menor (kernel/filtro) que a AlexNet (por exemplo, na primeira camada de convolução (ZEILER; FERGUS, 2014), a rede usava kernels de tamanho 7×7 , enquanto a AlexNet usava *kernels* de tamanho 11×11), a rede VGG usou *kernels* muito pequenos (3×3) em todas as suas camadas de convolução.

Há três aspectos importantes que devem ser esclarecidos para justificar porque os pequenos *kernels* permitiram à rede VGG dominar a maioria dos modelos anteriores. Primeiro, em uma convolução, um determinado ponto no mapa de ativação é influenciado apenas por uma sub-região de entrada conectada, por exemplo, uma convolução de 3×3 é influenciada por uma janela de entrada 3×3 . Além disso, se uma segunda convolução de 3×3 for realizada, mas agora sobre o primeiro mapa de ativação, cada ponto do novo mapa de ativação será influenciado por uma janela de 3×3 do primeiro. No entanto, como cada ponto do primeiro mapa de ativação combina uma informação de 3×3 da entrada, uma combinação de 3×3 desses primeiros pontos de ativação é influenciada por uma informação de 5×5 da entrada. Portanto, se esse processo for repetido mais uma vez, cada ponto de saída da pilha de convolução 3×3 de três camadas será influenciado por uma sub-região de 7×7 da entrada. Em outras palavras, uma pilha de convolução 3×3 de três camadas tem o mesmo campo receptivo efetivo de uma única convolução 7×7 .

Segundo, uma pilha de convolução de 3×3 de três camadas possui menos parâmetros que uma única camada convolucional de 7×7 . Terceiro, como cada camada tem sua não-linearidade, uma pilha de camadas de convolucionais tem mais poder discriminativo do que apenas uma camada. Portanto, as redes VGG tornaram-se bastante populares não apenas por causa de seus bons resultados no 2014 ILSVRC (RUSSAKOVSKY *et al.*, 2015), mas também por sua arquitetura uniforme.

3.3 Arquiteturas Inception

Desde a LeNet5 (LECUN *et al.*, 1998), os elementos básicos das arquiteturas neurais profundas são os mesmos (camadas convolucionais, camadas de *pooling* e camadas totalmente conectadas), mas isso mudou quando a equipe do *Google Brain* lançou o GoogleNet (SZEGEDY *et al.*, 2015). Inception, como também é chamado o GoogleNet, define resultados de ponta na tarefa de detecção 2014 ILSVRC (RUSSAKOVSKY *et al.*, 2015), atingindo uma taxa de erro de 6,67%.

Além disso, entre as contribuições dos modelos Inception ao campo de pesquisa de aprendizagem profunda estão o módulo *Inception*, *Batch-Normalization* e Convoluções Assimétricas.

3.3.1 *Módulo Inception*

Projetar uma arquitetura CNN geralmente consiste em escolher quais operações usar (convolução 1×1 , convolução 3×3 , convolução 5×5 ou *pooling*) em cada camada. Neste módulo foi proposto a execução de todas as operações de uma só vez e concatenar suas saídas, aproveitando a extração de recursos em vários níveis.

A principal desvantagem deste Módulo *Inception* simples é que, depois de algumas camadas como essa, a quantidade de parâmetros adicionados à rede provavelmente a tornaria excessiva durante o treinamento.

No entanto, para abordar a ineficiência desse parâmetro, o *Google Brain* realizou uma convolução de 1×1 para reduzir a dimensionalidade. Essa adaptação simples preservou a maior parte da precisão da abordagem paralela e reduziu significativamente a quantidade de parâmetros livres, e desta forma, removendo gargalos computacionais.

3.3.2 *Camada Batch-Normalization*

Um dos problemas mais desafiadores na otimização de redes profundas é o Deslocamento Covariado Interno (*Internal Covariate Shift*) (IOFFE; SZEGEDY, 2015), que é, basicamente, uma alteração na distribuição de entradas de camadas intermediárias no momento do treinamento. Isso acontece porque as entradas de uma determinada camada são afetadas por todos os pesos das camadas anteriores, que geralmente mudam com a otimização. Além disso, como os pesos das camadas precisam se adaptar a uma nova distribuição de entrada a cada atualização de peso, um efeito negativo aparece e a convergência do modelo se torna mais lenta.

Para resolver o problema de *Internal Covariate Shift*, Ioffe e Szegedy (2015) propuseram a normalização em lote *batch-normalization*, um novo mecanismo que incorpora ativações de normalização na própria arquitetura. Com apenas dois parâmetros extras por ativação, o *batch-normalization* normaliza as saídas do produto interno entre pesos e entradas de uma determinada camada, com base em sua média e desvio padrão. Isso é equivalente a “limpar” os dados, fazendo uma distribuição centralizada em zero e normalizando seu intervalo ao longo do lote. Além disso, entre os benefícios do *batch-normalization* estão: a camada após um *batch-normalization* se concentra em como melhor associar insumos; o *batch-normalization* tem um poderoso efeito de regularização e reduz ou elimina a necessidade de *dropout*.

3.3.3 Convoluções Assimétricas

Estendendo o conceito de Convoluções Fatoradas, aplicado anteriormente no artigo VGG (SIMONYAN; ZISSERMAN, 2014), e também com base nas Flattened CNN (JIN *et al.*, 2014), a equipe do *Google Brain* usou Convoluções Assimétricas para propor uma nova versão dos Módulos *Inception*. Com o objetivo de melhorar a eficiência dos parâmetros do Módulo *Inception*, eles percebem que, ao invés de executar uma convolução $n \times n$, seria melhor aplicar uma convolução $n \times 1$ seguida por uma convolução $1 \times n$, assim, fatorando-a em duas convoluções assimétricas.

3.4 Arquiteturas ResNet

Apesar das indicações do VGG, apontando para uma correlação positiva entre a profundidade e a precisão da rede, outras experiências, realizadas em um artigo da Rede Neural Residual (ResNet) (HE *et al.*, 2016a), não suportam essa visão, pelo menos para CNN clássicas (redes compostas apenas por camadas de convoluções, camadas de *pooling* e camadas totalmente conectadas).

Os autores da ResNet argumentam que CNN clássicas muito profundas são mais difíceis de otimizar, principalmente devido ao problema do gradiente de fuga (BENGIO *et al.*, 1994). Na prática, uma rede de 18 camadas apresenta erros de teste e treinamento mais altos do que uma rede de 34 camadas, mesmo que essas 16 camadas adicionais executem apenas um mapeamento de identidade.

Para superar esse problema e inspirados no conceito de “skip connections” das Redes Neurais Recorrentes, os autores da ResNet introduziram um novo bloco de construção de redes neurais profundas, o Bloco Residual.

3.4.1 Bloco Residual

Em um bloco residual, a entrada flui para duas ramificações, uma conecta a entrada ao estágio de mesclagem e a outra executa uma pilha de camadas de convoluções (geralmente 2 ou 3) preservando as dimensões da entrada (por exemplo, se a entrada tiver C canais, então, C filtros são aplicados) e envia os resultados para o estágio de mesclagem. Finalmente, as saídas de ambos os ramos são somadas. Observe que a última camada de convolução da segunda ramificação não possui não-linearidades após as operações de convoluções. Essas funções de

ativação ausentes são aplicadas à saída do estágio de mesclagem, compondo a saída do bloco residual.

Com essa estratégia simples, a ResNet definiu resultados de última geração em tarefas de detecção, localização e segmentação de objetos no ILSVRC (RUSSAKOVSKY *et al.*, 2015) e COCO 2015 (LIN *et al.*, 2014), atingindo um erro de classificação entre os 5 melhores, com taxa de erro de 3,57% (ILSVRC) e superando até o desempenho em nível humano no conjunto de dados *ImageNet*.

3.5 Arquiteturas Inception-ResNet

Em 2016, menos de um ano depois da ResNet, a equipe do Google Brain realizou experimentos de design de arquiteturas combinando as ideias de Conexões Residuais (HE *et al.*, 2016a) com o Módulo Inception (SZEGEDY *et al.*, 2015). Entre esses experimentos está o modelo Inception-ResNet (SZEGEDY *et al.*, 2017), que superou os resultados das arquiteturas ResNet e Inception.

3.5.1 Módulo Inception com Conexões Residuais

A única diferença entre uma combinação direta de um Módulo Inception com Conexões Residuais é uma camada de escala na saída do Módulo Inception. Aparentemente, sem esse estágio de dimensionamento, a rede começa, após algumas milhares de iterações, a produzir apenas zeros. Os autores realizaram várias experiências para lidar com isso, mas apenas diminuindo, por um fator entre 0,1 e 0,3, a saída inicial antes de adicioná-las às entradas estabilizou o procedimento de otimização.

3.6 Arquitetura Xception

Em 2016, François Chollet lançou o documento *Extreme Inception* (Xception) (CHOLLET, 2017), investigando a hipótese de que as correlações entre canais podem ser totalmente dissociadas das correlações espaciais. Para isso, o autor propõe a camada de Convolução Separável por Profundidade. Vale ressaltar que, com 36 camadas de Convolução Separáveis por Profundidade estruturadas em 14 módulos (a maioria delas com Conexões Residuais), a arquitetura Xception alcançou bons resultados no conjunto de dados JFT (HINTON *et al.*, 2015), um banco de dados de 350 milhões de imagens com 17.000 rótulos possíveis.

3.6.1 *Convoluções Separáveis por Profundidade*

A ideia por trás de uma camada de convolução 2D são recursos aprendidos diretamente de um espaço 3D (largura, altura e canais). Assim, uma convolução mapeia correlações entre canais e espaciais simultaneamente. Além disso, em um Módulo *Inception*, uma convolução é fatorada em um conjunto de convoluções 1×1 , que são influenciadas especificamente por correlações entre canais, seguidas por convoluções regulares (por exemplo, convoluções 3×3 ou 5×5), que mapeiam as correlações espaciais e os canais cruzados.

No entanto, em Convolução Separável por Profundidade, primeiro é realizada uma convolução em canal, que mapeia apenas correlações espaciais; então, um conjunto de convoluções é calculado, mapeando as correlações entre canais e, em seguida, finalmente, todas as saídas de convoluções são concatenadas em profundidade.

Portanto, a correlação espacial é totalmente dissociada das correlações entre canais. Entre os benefícios da Convolução Separável por Profundidade sobre o Módulo *Inception* clássico estão uma redução no número de parâmetros livres e uma execução mais eficiente. Essas duas características são valiosas para aplicativos móveis ou mesmo embarcados.

3.7 **Arquitetura MobileNet**

Em 2017, os pesquisadores do Google projetaram e lançaram uma arquitetura *mobile-first* chamada MobileNet (HOWARD *et al.*, 2017). Essa rede de 28 camadas é essencialmente (exceto a primeira camada) construída em convoluções separáveis por profundidade, que foram introduzidas por Howard *et al.* (2017) e alcançaram, em sua configuração de linha de base, 99,1% de acurácia de detecção no banco de dados *ImageNet* com 87,7% menos de parâmetros e 88,6% menos de operações.

A abordagem MobileNet distingue-se do modelo Xception pela introdução de dois hiperparâmetros (multiplicador de largura, multiplicador de resolução), que permitem que o modelo corresponda, de forma eficiente, às restrições de hardware dos sistemas móveis e embarcados.

3.7.1 *Multiplicadores de Largura*

Para criar arquiteturas flexíveis e eficientes, os autores do MobileNet propõem um hiperparâmetro α , também chamado de “multiplicador de largura”. Nesta arquitetura, esse

parâmetro α é um fator de redução no número de convoluções em canais, em camadas de convolução separáveis por profundidade. Além disso, α afeta quase quadraticamente o número de parâmetros e o custo computacional do modelo. Observe que $\alpha \in (0, 1]$, em que 1 é a configuração da linha de base da MobileNet.

3.7.2 Multiplicadores de Resolução

O segundo hiperparâmetro proposto é o multiplicador de resolução ρ . Na prática, esse parâmetro ρ adapta a arquitetura da rede para processar diferentes resoluções de entrada (largura e altura). Por exemplo, se ρ for igual a 1, a entrada de rede será uma imagem 224×224 , se ρ for igual a 0,75, a entrada será uma imagem 168×168 e assim por diante. Observe que, como $\alpha, \rho \in (0, 1]$ (sendo $\rho = 1$ a linha de base) e ρ também têm um efeito quadraticamente sobre o custo computacional.

Assim, o multiplicador de largura e os hiperparâmetros de resolução oferecem uma troca útil entre tamanho, precisão e latência, ao direcionar um sistema com limitações de hardware.

3.8 Arquiteturas DenseNet

A Rede Convolutiva Densa (DenseNet), proposta por Huang *et al.* (2017), estendeu o conceito de conexões mais curtas, anteriormente explorado por Szegedy *et al.* (2015), compondo o que é chamado de Blocos Densos.

3.8.1 Blocos Densos

Existem duas diferenças principais entre as arquiteturas DenseNet e ResNet. Primeiro, em vez de executar conexões mais curtas entre as camadas adjacentes, a DenseNet conecta cada camada a todas as outras camadas com a mesma resolução de entrada (largura e altura). Esse conjunto de camadas interconectadas é denominado Bloco Denso. Segundo, toda camada DenseNet executa uma concatenação profunda das saídas anteriores, ao invés de somar.

Assim, como resultado dessa estratégia, a DenseNet supera ligeiramente os resultados de acurácia do ResNet na tarefa de detecção no conjunto de dados ImageNet, exigindo aproximadamente 55% menos parâmetros e 50% menos operações que a ResNet.

3.9 Arquiteturas NASNet

Com base no pressuposto de que projetar arquiteturas de classificação de imagem precisas e eficientes geralmente é um processo complexo de engenharia de arquitetura, Zoph e Le (2016) explorou a hipótese possível de aprender com eficiência uma arquitetura de modelo diretamente a partir de um conjunto de dados de interesse. Inspirada nos experimentos de *Neural Architecture Search* (NAS), a equipe do *Google Brain* propôs um novo paradigma de design de arquiteturas convolucionais, o espaço de busca da NASNet.

3.9.1 NASNet Search Space

Esse paradigma restringe o amplo espaço de pesquisa, combinando vários blocos de construção (por exemplo, identidade, convolução 1×3 e 3×1 , convolução 1×7 e 7×1 , *max pooling* 5×5 , convolução 1×1 , convolução 3×3 , convolução separáveis por profundidade 3×3 etc.), para formar uma rede inteira, em uma menor, procurando apenas dois módulos arquitetura (célula normal e célula de redução).

Além disso, para acelerar esse processo de pesquisa, a NASNet se baseia na suposição de que é possível transferir um módulo aprendido em um pequeno conjunto de dados, por exemplo, CIFAR-10, para um maior, como a ImageNet.

Portanto, a NASNet alcançou resultados de acurácia de detecção satisfatórios no conjunto de dados ImageNet. Além disso, em sua versão lite, o NASNet corresponde à eficiência dos parâmetros do MobileNet, porém, é mais preciso e mais rápido. No entanto, vale ressaltar que, apesar de usar 500 GPUs Nvidia K40, a pesquisa de arquitetura mais rápida levou 4 dias.

3.10 Transfer Learning

O *Transfer Learning* (aprendizagem por transferência) é um método que utiliza o conhecimento adquirido por uma CNN a partir de um problema específico para resolver uma tarefa distinta. Esse conhecimento transferido é usado em um novo conjunto de dados, cujo tamanho geralmente é menor do que o adequado para treinar uma CNN do zero (PAN; YANG, 2009).

No aprendizado profundo, esse método requer um treinamento inicial de um CNN para uma determinada tarefa, usando grandes conjuntos de dados. A disponibilidade de um conjunto de dados considerável é o principal fator para garantir o sucesso do método, uma vez

que a CNN pode aprender a extrair as características mais significativas de uma amostra. A CNN é considerada adequada para aprendizagem por transferência se for capaz de extrair os recursos de imagem mais importantes (WEISS *et al.*, 2016).

Então, no *Transfer Learning*, a CNN é utilizada para analisar um novo conjunto de dados de natureza diferente e extrair suas características de acordo com o conhecimento adquirido no primeiro treinamento. Uma estratégia comum para explorar os recursos da CNN pré-treinada é chamada de extração de características via *Transfer Learning* (HUH *et al.*, 2016).

A abordagem *Transfer Learning* é usada principalmente para contornar os custos computacionais de treinamento de uma rede do zero ou para manter o extrator de características treinado durante a primeira tarefa. Em aplicações médicas, a prática mais aceita de *Transfer Learning* é utilizar as CNN que alcançaram os melhores resultados no ILSVRC (RUSSAKOVSKY *et al.*, 2015), em que avaliam algoritmos para detecção e classificação de objetos em grandes escalas. O uso de grandes conjuntos de dados para o treinamento inicial da rede permite alto desempenho em conjuntos de dados menores. Este desempenho está ligado a vários parâmetros de extração que normalmente não são permitidos, pois causam *overfitting* da rede (NÓBREGA *et al.*, 2018). Dito isso, a extração de características realizada com transferência de aprendizagem permite que um grande número de características sejam extraídas generalizando o problema e evitando ajustes excessivos (ZHAO, 2017).

O uso de *Transfer Learning* também permite o uso de sistemas de Internet das Coisas (*Internet of Things*) (IoT) para classificar imagens médicas. Por exemplo, Jr *et al.* (2019) propuseram um sistema IoT para detectar acidente vascular cerebral em imagens de tomografia computadorizada. Rodrigues *et al.* (2019) usaram o sistema proposto por Jr *et al.* (2019) para classificar sinais de eletroencefalograma.

Visto que diferentes configurações de CNN podem produzir os mais diversos resultados, diferentes arquiteturas de CNN devem ser testadas para encontrar um extrator de características adequado para uma tarefa específica. Na Tabela 1, são apresentadas as arquiteturas utilizadas nesta pesquisa e seus respectivos destaques. Essas CNN foram escolhidas devido à diversidade em suas estruturas, conforme apresentado acima, e seus excelentes resultados no ILSVRC.

Por fim, o método de *Transfer Learning* é usado na etapa de extração de características das imagens histológicas de CRC. O processo é detalhado na Seção 4.2.

Tabela 1 – Arquiteturas de CNN usadas como extratores de características e seus respectivos destaques.

Arquiteturas	Destaques
VGG (SIMONYAN; ZISSERMAN, 2014)	Maior poder discriminativo devido ao aumento da profundidade da CNN, possibilitado pelo uso de filtros menores
Inception (SZEGEDY <i>et al.</i> , 2015)	Execução de múltiplas convoluções paralelas em blocos, melhorando o desempenho da rede com um ligeiro aumento no custo computacional
ResNet (HE <i>et al.</i> , 2016a)	Uso de “blocos residuais”, que aprendem funções residuais de entradas de camadas, permitindo maior precisão conforme a profundidade da rede é aumentada
NASNet (ZOPH; LE, 2016)	Construção de arquiteturas que são aprendidas diretamente de um conjunto de dados de interesse
Xception (CHOLLET, 2017)	Substituição de “Módulos Inception” por “Convolução Separável por Profundidade”
MobileNet (HOWARD <i>et al.</i> , 2017)	Introdução de dois hiperparâmetros que permitem a adaptação do tamanho do modelo de acordo com as restrições do problema
DenseNet (HUANG <i>et al.</i> , 2017)	Conexão de cada camada com todas as outras camadas em um modo <i>feed-forward</i> , fortalecendo a propagação de características

Fonte: Elaborado pelo autor.

4 METODOLOGIA

Este capítulo descreve a metodologia adotada para o desenvolvimento da abordagem proposta, no qual consiste em três etapas principais: i) extração de características, ii) classificação e iii) avaliação dos resultados. Na primeira etapa são extraídas as características do conjunto de dados original por meio da extração profunda de recursos, utilizando a técnica *Transfer Learning*, a qual resulta na criação de subconjuntos de dados. Em seguida, esses sub-conjuntos de dados são utilizados como entrada para os classificadores. Finalmente, os resultados são avaliados, e então, é escolhida a melhor combinação de extrator-classificador de características para classificar as imagens histológicas do CRC. Uma visão geral da abordagem proposta é mostrada na Figura 4.

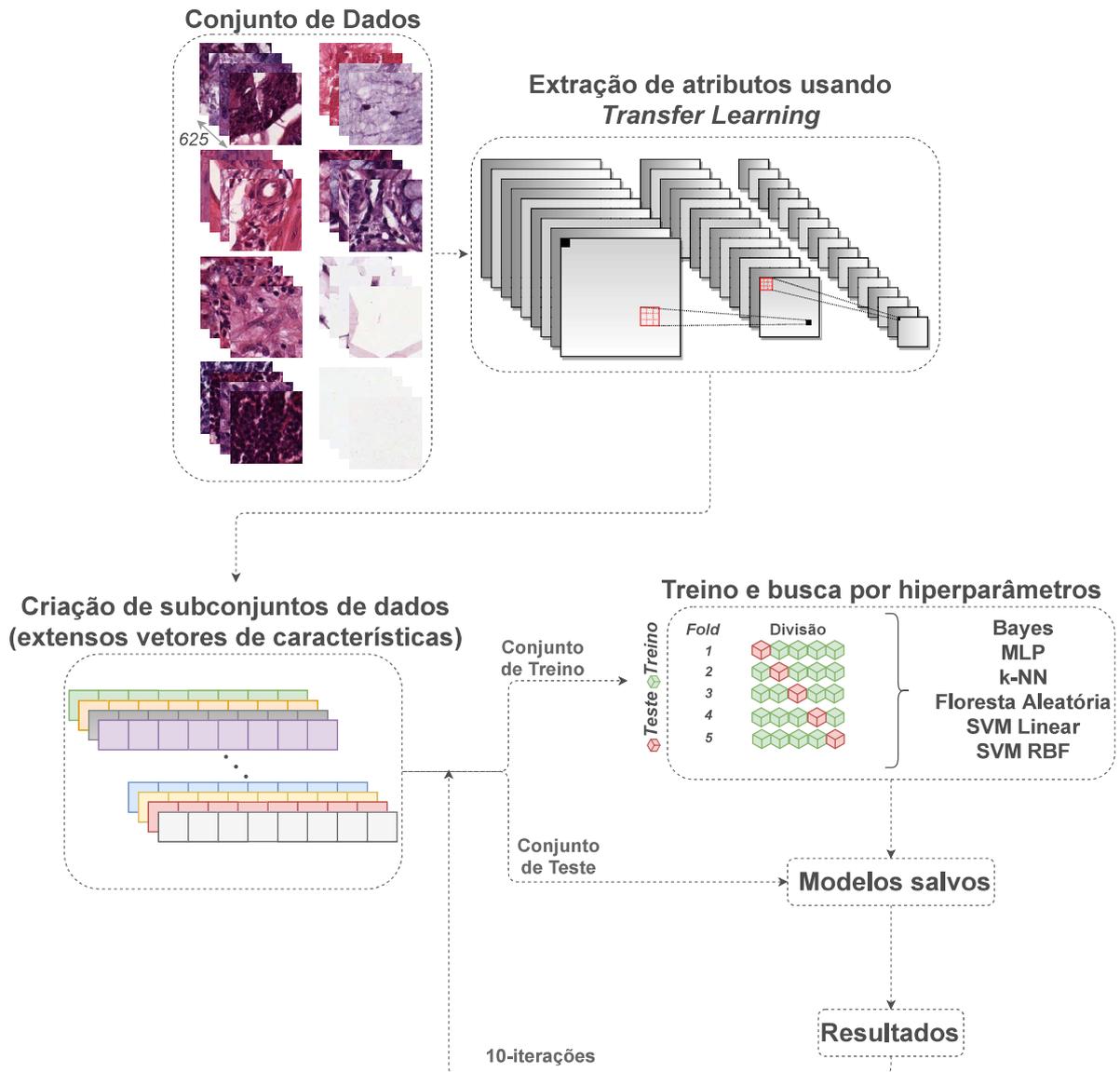
Este capítulo está dividido da seguinte forma: a Seção 4.1 descreve o conjunto de dados, a Seção 4.2 detalha a etapa de extração de características, a Seção 4.3 expõe as etapas de classificação, e a Seção 4.4 apresenta as métricas empregadas para avaliar os resultados da classificação das imagens histológicas de CRC.

4.1 Conjunto de dados

O conjunto de dados usado nesta abordagem corresponde a uma coleção de imagens de exames histopatológicos. O conjunto de dados, chamado de *Colorectal Cancer Histology*, fornece uma coleção de texturas na histologia do CRC, e foi disponibilizada publicamente em 26 de maio de 2016 (KATHER *et al.*, 2016). As imagens representam uma coleção de texturas encontradas em imagens histológicas de CRC humano. O conjunto de dados contém 5.000 imagens histológicas divididas em oito classes (KATHER *et al.*, 2016), que estão listados abaixo:

1. Epitélio tumoral
2. Estroma simples: composição homogênea do estroma tumoral, extratumoral e músculo liso
3. Estroma complexo: composição de células tumorais individuais e células do sistema imunológico
4. Células imunes: abrange conglomerados de células imunes e folículos linfóides sub-mucosos
5. Detritos: cobre regiões com necrose, muco e hemorragia
6. Glândulas mucosas normais
7. Tecido adiposo

Figura 4 – Metodologia da abordagem proposta.



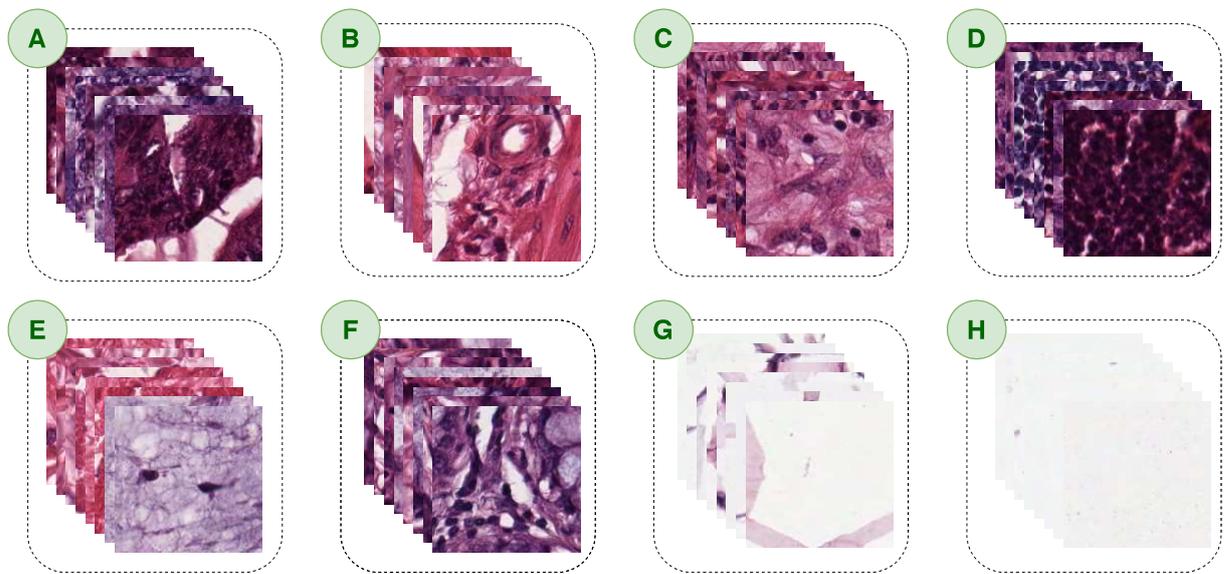
Fonte: Elaborado pelo autor.

8. Fundo: nenhum tecido.

Todas as imagens são distribuídas uniformemente pelas classes, correspondendo a 625 imagens de 150×150 pixels ($74 \times 74 \mu m$) para cada classe. As imagens foram obtidas em um *Aperio ScanScope*, que tem capacidade de aumento de 20x. As imagens foram disponibilizadas pelo *Institute of Pathology at the University Medical Center Mannheim* (Heidelberg University, Mannheim, Alemanha), que garante e protege a integridade da origem do exame/paciente. Todas as amostras foram coletadas e aprovadas de acordo com o Comitê de Ética Médica II da *University Medical Center Mannheim* (Heidelberg University, Alemanha) e pela Declaração de Helsinki (KATHER *et al.*, 2016).

Na Figura 5, é possível notar a variância encontrada no mesmo tipo de tecido e a semelhança entre amostras de diferentes classes. Essa característica específica justifica a complexidade da classificação, o que a torna suscetível à subjetividade e pode resultar em divergências entre especialistas.

Figura 5 – Amostras de cada classe do conjunto de dados *Colorectal Cancer Histology*. Da esquerda para a direita, conjuntos de oito imagens aleatórias para as classes (A) epitélio tumoral, (B) estroma simples, (C) estroma complexo, (D) células imunes, (E) detritos e muco, (F) glândula mucosa, (G) tecido adiposo, e (H) fundo.



Fonte: Elaborado pelo autor.

4.2 Extração de características

Neste trabalho, é proposto o uso de CNN para extrair características de imagens histológicas de CRC humano. É utilizado uma abordagem *Transfer Learning* padrão para transformar uma CNN em um extrator de características profundo (ORENSTEIN; BEIJBOM, 2017; PAUL *et al.*, 2016). A técnica de *Transfer Learning* pode reduzir o tempo de treinamento necessário para obter uma acurácia semelhante a uma classificação da arquitetura CNN (TANG *et al.*, 2019), sem necessidade de uma Unidade de Processamento Gráfico (*Graphical Processing Unit*) (GPU). Além disso, não requer dados de treinamento em grande escala (PAN; YANG, 2009).

Esta abordagem é alcançada primeiramente treinando uma arquitetura CNN em um grande conjunto de dados, onde a rede aprende a extrair características críticas das imagens.

Neste método, foram utilizados os pesos pré-treinados no conjunto de dados *ImageNet*. Em seguida, foram removidas as camadas totalmente conectadas da CNN, responsáveis por classificar os atributos, e as camadas convolucionais e de *pooling* foram preservadas. Como resultado, a saída dessa CNN adaptada torna-se um conjunto de atributos extraídos da imagem de entrada.

4.2.1 CNN como extrator de características

Esse tópico detalha o processo de construção e inicialização das arquiteturas, seu treinamento (com o conjunto de dados de referência), pré-processamento, e suas transformações em extratores de características.

4.2.1.1 Construção e inicialização das arquiteturas

Nos últimos anos, especialmente após 2012, o desempenho na tarefa de detecção de objetos aumentou significativamente com o desenvolvimento de várias arquiteturas de CNN. No entanto, como é inviável avaliar a hipótese explorada neste trabalho em todas as arquiteturas elaboradas pela comunidade científica, sete arquiteturas respeitáveis (muitas delas vencedoras de desafios complexos como ILSVRC (RUSSAKOVSKY *et al.*, 2015) e COCO (LIN *et al.*, 2014)) foram selecionadas para realizar a extração de atributos.

Mais especificamente, foram utilizadas as seguintes configurações: VGG16 (SIMONYAN; ZISSERMAN, 2014), VGG19 (SIMONYAN; ZISSERMAN, 2014), InceptionV3 (SZEGEDY *et al.*, 2015), InceptionResNetV2 (SZEGEDY *et al.*, 2015), ResNet50 (HE *et al.*, 2016a), ResNet50V2 (HE *et al.*, 2016b), ResNet101 (HE *et al.*, 2016a), ResNet101V2 (HE *et al.*, 2016b), ResNet152 (HE *et al.*, 2016a), ResNet152V2 (HE *et al.*, 2016b), NASNetLarge (ZOPH; LE, 2016), NASNetMobile (ZOPH; LE, 2016), Xception (CHOLLET, 2017), MobileNet (HOWARD *et al.*, 2017), MobileNetV2 (SANDLER *et al.*, 2018), DenseNet121, DenseNet169 and DenseNet201 (HUANG *et al.*, 2017). A estrutura principal e os parâmetros das redes dos artigos originais foram preservados. Manter as características dessas redes permite o uso de um extrator de características robusto, inicialmente usado para um desafio mais complexo, para resolver uma tarefa menor.

4.2.1.2 *Treinamento das Arquiteturas*

A segunda etapa desta seção consiste em treinar as 18 configurações de CNN contempladas neste trabalho no conjunto de dados de referência, ou seja, o conjunto de dados do qual o “conhecimento” é transferido.

Com base na premissa de que as características/padrões aprendidos por uma CNN para discriminar um conjunto de classes são mais genéricos do que aqueles aprendidos a discriminar apenas um subconjunto dessas classes, o conjunto de dados usado nesta etapa devem ter o maior número possível de classes. Além disso, como todas as sete arquiteturas foram elaboradas com o objetivo de solucionar o problema representado pelo conjunto de dados *ImageNet*, que consiste em 1,2 milhão de imagens não médicas e divididas em 1000 categorias, as metodologias de treinamento usadas por cada uma dessas arquiteturas são documentadas detalhadamente em seus respectivos artigos. Nessa etapa, os pesos pré-treinados no conjunto de dados *ImageNet* foram usados.

4.2.1.3 *Pré-processamento*

Como etapa de pré-processamento, foi realizado um redimensionamento das imagens para o tamanho de entrada de cada CNN. O tamanho de imagem de entrada para cada configuração de CNN pode ser encontrada na Tabela 2. Esse redimensionamento foi executado usando a interpolação do vizinho mais próximo disponível na biblioteca *OpenCV*, o qual não tem um custo computacional significativo.

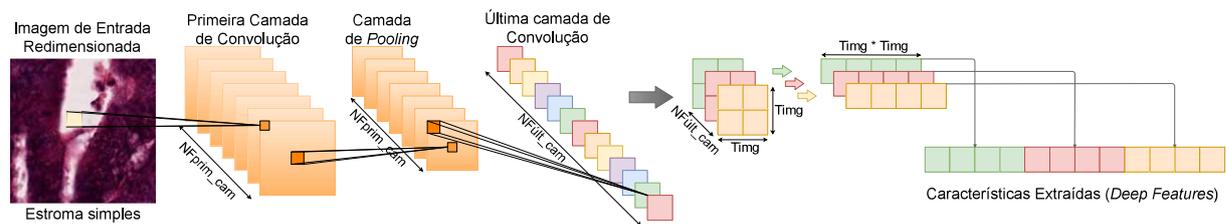
4.2.1.4 *Convertendo CNN em extratores de características*

Na última etapa desta seção, as CNN treinadas no conjunto de dados de referência são convertidas em extratores de recursos. Entretanto, para isso, é necessário entender as quatro transformações realizadas por essas redes neurais.

Primeiro, a imagem de entrada da CNN, ou seja, a imagem redimensionada a ser processada, é submetida inicialmente a uma sequência de transformações não lineares, definidas de acordo com a arquitetura utilizada. Mais especificamente, esse processo converte a imagem de entrada em um conjunto de pequenas matrizes (em nenhuma das arquiteturas usadas neste trabalho, a resolução dessas matrizes é maior do que $11 \times 11 \times 1$). Posteriormente, cada uma dessas matrizes é redimensionada para um vetor unidimensional. Então, o conjunto desses

vetores é concatenado, formando assim um único vetor. Este vetor, por sua vez, pode ser interpretado como os atributos extraídos pela CNN. O processo descrito é ilustrado na Figura 6, onde NF_{prim_cam} simboliza a quantidade de filtros aplicados nas primeiras camadas, $NF_{últ_cam}$ representa a quantidade de filtros aplicados na última camada e T_{img} é o tamanho da imagem de entrada após as transformações de todas as camadas da CNN.

Figura 6 – A imagem de entrada é submetida a uma sequência de camadas convolucionais e sub-amostragem (*pooling*). Os resultados são redimensionados e concatenados em uma única lista, chamada de *deep features*.



Fonte: Elaborado pelo autor.

Por fim, esses vetores de atributos são submetidos a uma ou mais camadas totalmente conectadas. Em geral, esse processo se assemelha ao processamento realizado pelos classificadores Perceptron ou MLP. Deve-se notar que as funções de ativação usadas nessas camadas são *ReLU* ou *Sigmoid*, para as primeiras camadas, e *Softmax*, para a última (se a rede tiver apenas uma camada totalmente conectada, a função de ativação usada é a *Softmax*). No entanto, as últimas camadas totalmente conectadas (usadas para treinar a rede no banco de dados de referência) são removidas do modelo.

Com isso, quando uma imagem é submetida a essa arquitetura modificada, o resultado não pode mais ser interpretado como um conjunto de probabilidades de uma imagem de entrada pertencente a uma determinada classe do conjunto de dados de referência (*ImageNet*), mas como um vetor de informação, que seria usado por camadas totalmente conectadas para compor as probabilidades de classe.

Então, foi utilizado o conhecimento adquirido pela CNN adaptada para extrair recursos das imagens pré-processadas do conjunto de dados *Colorectal Cancer Histology* e construir novos subconjuntos. Como essa CNN não tem mais camadas totalmente conectadas, pode-se escolher um método de aprendizado de máquina para classificar as características.

A Tabela 2 apresenta as diferentes configurações, suas arquiteturas de base, o tamanho da imagem de entrada necessário para cada configuração e o número resultante de recursos

extraídos de uma única imagem. Por exemplo, a configuração InceptionResNetV2 deriva das arquiteturas Inception e ResNet.

Tabela 2 – Arquiteturas de CNN, as configurações utilizadas, o tamanho de imagem de entrada, e o número de atributos extraídos.

Arquitetura	Configurações	Tamanho da imagem de entrada (em pixels)	Quantidade de atributos extraídos
VGG (SIMONYAN; ZISSERMAN, 2014)	VGG16	224 × 224	512
	VGG19	224 × 224	512
Inception (SZEGEDY <i>et al.</i> , 2015)	InceptionV3	299 × 299	2048
	InceptionResNetV2	299 × 299	1536
ResNet (HE <i>et al.</i> , 2016a)	ResNet50	224 × 224	2048
	ResNet50V2 (HE <i>et al.</i> , 2016b)	224 × 224	2048
	ResNet101	224 × 224	2048
	ResNet101V2 (HE <i>et al.</i> , 2016b)	224 × 224	2048
	ResNet152	224 × 224	2048
NASNet (ZOPH; LE, 2016)	ResNet152V2 (HE <i>et al.</i> , 2016b)	224 × 224	2048
	NASNetLarge	331 × 331	4032
Xception (CHOLLET, 2017)	NASNetMobile	224 × 224	1056
	Xception	299 × 299	2048
MobileNet (HOWARD <i>et al.</i> , 2017)	MobileNet	224 × 224	1024
	MobileNetV2 (SANDLER <i>et al.</i> , 2018)	224 × 224	1280
DenseNet (HUANG <i>et al.</i> , 2017)	DenseNet121	224 × 224	1024
	DenseNet169	224 × 224	1664
	DenseNet201	224 × 224	1920

Fonte: Elaborado pelo autor.

4.3 Classificação

Para atribuir cada imagem a uma das oito classes, foram selecionados métodos de aprendizado de máquina amplamente usados na literatura: Bayes (ZAW *et al.*, 2019; WU *et al.*, 2017; SINGH *et al.*, 2019), Floresta Aleatória (WANG, 2016; HEBBAR *et al.*, 2019; ZHANG *et al.*, 2019), MLP (GAO *et al.*, 2018; WAN *et al.*, 2018; DINO; ABDULRAZZAQ, 2019), k-NN (ZHANG *et al.*, 2017; XING; BEI, 2019; ZHU *et al.*, 2018), e SVM (ZHANG *et al.*, 2018; LOCONSOLE *et al.*, 2018; CHEN; ZHANG, 2019). No classificador SVM, foram considerados os *kernels* linear e função de base radial (*Radial Basis Function*, RBF). É importante ressaltar que esses classificadores são de diferentes tipos: k-NN é baseado em instância, Floresta Aleatória é baseado no método de árvore de decisão, MLP é baseado em redes neurais, SVM é baseado em

encontrar um hiperplano ótimo e Bayes é baseado em probabilidade e estatísticas. Nesta etapa, cada sub-conjunto de dados é composto pelos recursos extraídos pelas CNN.

Uma “busca” por hiperparâmetros foi realizada nos classificadores. Foi definido um intervalo de “busca” e utilizamos os métodos de pesquisa em grade (*GridSearch*) e pesquisa aleatória (*RandomSearch*), uma vez que são semi-automáticos. Foi realizado um *GridSearch* no número de vizinhos do classificador k-NN. Enquanto nos classificadores MLP, Floresta Aleatória e SVM, foi realizado um *RandomSearch* por meio de 20 iterações e validação cruzada de *5-folds*. A Tabela 3 apresenta os valores considerados para a “busca” de hiperparâmetros. O número de estimadores é o número de árvores de decisão. Em vez do procedimento de gradiente estocástico mais comumente usado, usamos um algoritmo de otimização chamado Adam (KINGMA; BA, 2014). No classificador SVM, o parâmetro C adiciona um custo para cada ponto de dados classificado incorretamente; e o parâmetro γ é usado para controlar a esfera de influência de um único ponto de treinamento.

Tabela 3 – Configurações utilizadas na “busca” pelos hiperparâmetros dos classificadores.

Classificador	Tipo de “busca”	Hiperparâmetro	Configuração
Bayes	-	-	Função densidade de Probabilidade Gaussiana
Floresta Aleatória	Random	número de estimadores	50 até 3000 em intervalos de 50
MLP	Random	neurônios na camada oculta	2 até 1000
k-NN	Grid	número de vizinhos	3, 5, 7, 9, 11, 13, 15
SVM (<i>kernel</i> linear)	Random	C	2^{-5} to 2^{15}
SVM (<i>kernel</i> RBF)	Random	C	2^{-5} até 2^{15}
		γ	2^{-15} até 2^3

Fonte: Elaborado pelo autor.

4.4 Métricas de Avaliação

A saída de cada combinação, produzida pela abordagem proposta, é avaliada usando a matriz de confusão e as métricas Acurácia e *F1-Score*. A matriz de confusão corresponde a uma ferramenta para avaliar algoritmos de classificação. O formato da matriz de confusão permite a visualização rápida e direta dos resultados obtidos para cada aula.

A matriz de confusão é composta por valores obtidos da classificação do conjunto de dados. Os valores são divididos entre as classes do conjunto de dados. Esses valores são

porcentagens do número de amostras previstas correta e incorretamente para cada classe do problema. A utilização dessa ferramenta permite relacionar as fragilidades do classificador, uma vez que fica evidente o percentual que está errado previsto entre cada classe.

A Tabela 4 apresenta a matriz de confusão que retrata as imagens de CRC previstas corretamente ou incorretamente por um método de classificação. Com base nos valores verdadeiro positivo (*True Positive*) (TP), falso positivo (*False Positive*) (FP), falso negativo (*False Negative*) (FN) e verdadeiro negativo (*True Negative*) (TN), as métricas foram calculadas expressando os valores da matriz de confusão. TP e FN são as amostras da classe desejada que estão corretamente e incorretamente classificadas, respectivamente. TN e FP são as amostras que não pertencem à classe desejada e estão corretamente e incorretamente classificadas, respectivamente.

Tabela 4 – Matriz de confusão para a classificação de imagens de CRC.

Classe Real	Classe Predita				
	0	1	2	...	7
0	TP	FN	FN	FN	FN
1	FP	TN	FN	FN	FN
2	FP	FN	TP	FN	FN
...	FP	FN	FN	TN	FN
7	FP	FN	FN	FN	TN

Fonte: Elaborado pelo autor.

A Acurácia demonstra o número de amostras previsto corretamente pelo modelo em relação ao número total de amostras. O *F1-Score* corresponde à quantidade que representa a média harmônica entre Precisão e Sensibilidade. A Precisão e a Sensibilidade são calculadas na geração dos resultados. Uma vez que alguns trabalhos usam a Taxa de Erro (TE) como uma métrica de avaliação, esta métrica é calculada apenas para comparação com outras abordagens na Seção 5.2. As equações 4.1, 4.2, 4.3, 4.4 e 4.5 representam os cálculos da Acurácia, Precisão, Sensibilidade, *F1-Score* e Taxa de Erro.

$$Acurácia = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

$$Precisão = \frac{TP}{TP + FP} \quad (4.2)$$

$$\text{Sensibilidade} = \frac{TP}{TP + FN} \quad (4.3)$$

$$F1\text{-score} = 2 \times \frac{\text{Sensibilidade} \times \text{Precisão}}{\text{Sensibilidade} + \text{Precisão}} \quad (4.4)$$

$$\text{Taxa de Erro (TE)} = 1 - \text{Acurácia} \quad (4.5)$$

No Capítulo 5, serão focados, principalmente, a Acurácia e o *F1-Score*. Uma alta acurácia indica um desempenho satisfatório e o *F1-Score* pode mostrar um equilíbrio entre a sensibilidade e a precisão das combinações. Assim, um método com alto *F1-Score* tende a classificar as regiões dos tecidos de forma mais uniforme, fazendo com que a combinação não resulte em um número significativo de falsos positivos e falsos negativos.

Além das métricas mencionadas acima, também foram analisados os tempos de treinamento e teste. O tempo decorrido desde o início do treinamento do classificador até o momento em que ele está pronto para classificar é definido como tempo de treinamento. O tempo de teste representa o tempo necessário para prever uma classe depois que o classificador recebe o vetor de características. Portanto, o tempo de treinamento é de suma importância durante a configuração do modelo. Depois que a configuração for concluída, o tempo de teste é mais crítico.

4.5 Validação cruzada

A validação cruzada (*cross-validation*) é uma técnica, baseada em estatística, bastante empregada para análise e comparação de métodos de classificação. Ela consiste na divisão dos dados em dois grupos diferentes, um grupo para a fase de treinamento e outro grupo para a fase de teste dos algoritmos (STONE, 1974).

Há dois tipos mais comuns de validação cruzada: *hold-out* e *k-fold*. O *hold-out* fraciona o conjunto de amostras em dois subconjuntos, em que um dos subconjuntos é dedicado para a etapa de treinamento e o outro para a etapa de validação. Neste trabalho, cada subconjunto de dados, os quais são formados pelos atributos resultantes após a etapa de extração de características, tiveram seus padrões divididos em dois conjuntos, onde oitenta por cento das

amostras foram destinadas para a fase de treinamento e o remanescente foi submetido para a fase de teste. Foram realizadas 10 iterações desse processo.

Quanto ao *k-fold*, ele efetua a partição das amostras em *k* subgrupos. São executadas *k* iterações para o treinamento e, em cada uma, um subconjunto é dedicado para a etapa de teste, enquanto os subconjuntos restantes formam o conjunto destinado para o treino, ou seja, são aplicados *k-folds* para a fase de treinamento e os outros *folds* são destinados para a fase de teste do algoritmo. Na abordagem proposta neste trabalho, para cada repetição do *hold-out*, foi considerada a validação cruzada com *5-folds* no conjunto de treinamento, na busca em grade ou randômica.

5 RESULTADOS E DISCUSSÕES

Os resultados desta dissertação são apresentados em duas etapas. Na primeira etapa (Seção 5.1), todas as combinações de arquiteturas CNN com os classificadores considerados tiveram suas métricas de avaliação comparadas. Nesta etapa, os tempos de processamento também são analisados. Na segunda etapa é realizada uma comparação entre esses resultados com aqueles obtidos a partir de estudos que utilizaram o mesmo conjunto de dados (Seção 5.2).

A metodologia aplicada foi implementada em um computador composto por um sistema operacional Ubuntu 16.04, com um processador Intel Xeon, 32GB de RAM, sem GPU. Em adição, foi utilizado a linguagem de programação Python 3.6, juntamente com as bibliotecas *Keras* v2.3.1, *Scikit-Learn* v0.21.0, e *OpenCV* v4.1.0.

5.1 Análise de Desempenho

Nesta seção, são apresentados os resultados dos $18 \times 6 = 108$ experimentos de classificação de imagens supervisionadas, sendo estes a combinação dos 18 extratores de características e os seis ¹ classificadores avaliados. Os resultados apresentam a média de uma validação cruzada de 10 iterações utilizando a técnica *hold-out*, onde 20% dos dados foram destinados a etapa de teste em cada iteração.

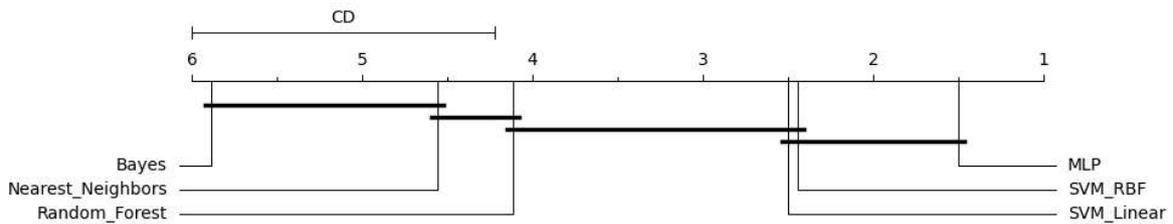
Foi aplicado o teste de Friedman na métrica acurácia dos resultados para testar se não há diferença estatística entre os resultados dos classificadores. Há um efeito significativo no nível $p < 0,05$ encontrado para os classificadores ($F = 73,63$, $p = 5,08 \times 10^{-9}$). Esse resultado é confirmado pelo teste post-hoc de Nemenyi ($CD = 1.77$), o qual é apresentado na Figura 7. Pode-se observar que há indicação consistente de diferenças estatísticas, principalmente, entre Bayes e os classificadores MLP, SVM (RBF) e SVM (Linear) para os 108 experimentos avaliados.

A Tabela 5 apresenta a Acurácia e o F1-Score que foram obtidos através da extração utilizando a técnica de *Transfer Learning* e a seguir classificação dos atributos. A Tabela 5 também mostra que as combinações da DenseNet com o SVM (RBF) têm as duas maiores acurácias, em que a DenseNet201 alcançou uma acurácia de 91,996% e a DenseNet169 atingiu 92,083%; essas combinações estão destacadas em azul.

A Tabela 5 também mostra que a metodologia proposta alcançou acurácia e *F1-Score*

¹ O classificador SVM com diferentes *kernels* foram considerados como diferentes classificadores.

Figura 7 – Resultado do teste de Nemenyi para os classificadores utilizados (Bayes, k-NN - *Nearest_Neighbors*, Floresta Aleatória - *Random_Forest*, MLP, SVM (RBF) e SVM (Linear)).



Fonte: Elaborado pelo autor.

de pelo menos 70% quando o classificador de Bayes é desconsiderado. Além disso, deve-se destacar os extratores: DenseNet201, MobileNet e VGG16. Esses extratores de características alcançaram uma acurácia e um *F1-Score* de pelo menos 86%, independentemente do classificador. O SVM (RBF) pode ser destacado entre os classificadores, pois atingiu métricas de pelo menos 79%, independentemente do extrator de recursos.

Outro ponto que pode ser observado na Tabela 5 é que as versões 2 (V2) da ResNet50, Resnet101, Resnet152 melhoraram na classificação das imagens de histologia do CRC. No entanto, as melhorias realizadas na MobileNet não afetaram os atributos extraídos usando a técnica *Transfer Learning*, e então classificados.

A Tabela 6 mostra os tempos de treino e teste para os 108 experimentos. Pode-se observar que o classificador Bayes alcançou os tempos de treinamento e teste mais rápidos quando combinado com quaisquer extratores de características. Isso é esperado, uma vez que o classificador Bayes exige apenas o cálculo da probabilidade de um conjunto de recursos ser uma classe específica com base no conhecimento prévio. O classificador MLP obteve os maiores tempos de treinamento devido ao seu número de neurônios na camada oculta. Na Tabela 6, também pode-se observar que o k-NN alcançou os tempos de teste mais lentos para todos os extratores de características. Isto ocorre devido ao alto número de características extraídos por cada configuração de CNN.

A DenseNet169, combinada com o SVM (RBF), atingiu um tempo de treinamento de 179 segundos e um tempo de teste de 0,765 ms. Além disso, a DenseNet201, combinada com o SVM (RBF), atingiu um tempo de treinamento de 265 segundos e um tempo de teste de 0,948 ms. Portanto, o tempo de teste de ambos os extratores de características combinados com o classificador é adequado para aplicações em tempo real.

A Figura 8 apresenta um resumo das melhores combinações de extratores de atributos

Tabela 5 – Acurácia e *F1-Score* obtidos por cada combinação de extrator e classificador de características. Seus respectivos desvios padrão acompanham todas as métricas.

Extrator de características	Classificador	Acurácia	F1-Score
ResNet101V2	Bayes	72,423±0,893	72,683±0,817
	MLP	87,506±0,989	87,548±0,964
	k-NN	80,244±1,333	80,360±1,354
	Floresta Aleatória	84,545±1,741	84,620±1,710
	SVM Linear	87,424±0,993	87,467±0,968
	SVM RBF	88,544±1,230	88,558±1,199
DenseNet121	Bayes	79,244±1,315	79,294±1,305
	MLP	88,320±1,149	88,323±1,170
	k-NN	83,952±1,924	84,359±1,866
	Floresta Aleatória	87,998±0,929	87,995±0,890
	SVM Linear	88,701±0,847	88,744±0,843
	SVM RBF	90,139±1,325	90,171±1,322
DenseNet201	Bayes	80,297±0,998	80,339±1,074
	MLP	89,718±1,143	89,735±1,140
	k-NN	86,564±1,492	86,809±1,453
	Floresta Aleatória	89,462±1,053	89,481±1,061
	SVM Linear	90,659±1,081	90,715±1,061
	SVM RBF	91,996±0,774	92,009±0,776
MobileNetV2	Bayes	75,858±1,543	75,908±1,590
	MLP	89,279±1,647	89,287±1,617
	k-NN	86,124±0,913	86,260±0,889
	Floresta Aleatória	84,523±1,541	84,529±1,538
	SVM Linear	88,200±1,768	88,219±1,736
	SVM RBF	90,601±1,589	90,612±1,577
ResNet50V2	Bayes	74,277±1,070	74,602±1,121
	MLP	88,622±1,401	88,618±1,406
	k-NN	82,440±1,076	82,454±1,154
	Floresta Aleatória	84,659±1,068	84,654±1,120
	SVM Linear	87,836±1,410	87,846±1,435
	SVM RBF	89,416±1,143	89,424±1,147
NASNetLarge	Bayes	74,799±0,892	75,137±0,914
	MLP	88,504±2,040	88,513±2,045
	k-NN	81,786±1,500	81,939±1,501
	Floresta Aleatória	84,986±1,870	85,047±1,922
	SVM Linear	87,763±1,701	87,819±1,705
	SVM RBF	89,044±1,651	89,081±1,664
InceptionResNetV2	Bayes	76,478±1,970	76,779±1,839
	MLP	88,659±1,207	88,698±1,179
	k-NN	82,119±1,747	82,317±1,723
	Floresta Aleatória	83,361±1,450	83,373±1,395
	SVM Linear	88,656±1,557	88,750±1,526
	SVM RBF	88,657±1,648	88,745±1,605

Fonte: Elaborado pelo autor.

e classificadores, que são a DenseNet201 combinada com o SVM (RBF) e a DenseNet169 combinada com o SVM (RBF). É perceptível que a combinação com DenseNet169 é ligeiramente melhor em acurácia e *F1-Score*, e seu tempo de teste também é inferior ao da DenseNet201. Os

Tabela 5 – (Cont.) Acurácia e *F1-Score* obtidos por cada combinação de extrator e classificador de características. Seus respectivos desvios padrão acompanham todas as métricas.

Extrator de características	Classificador	Acurácia	F1-Score
DenseNet169	Bayes	80,427±1,931	80,583±1,966
	MLP	90,947±1,365	90,978±1,354
	k-NN	84,983±0,631	85,348±0,585
	Floresta Aleatória	88,842±1,492	88,877±1,469
	SVM Linear	90,644±0,863	90,680±0,849
	SVM RBF	92,083±1,029	92,117±1,013
MobileNet	Bayes	80,548±2,392	80,621±2,308
	MLP	90,525±1,904	90,540±1,912
	k-NN	87,545±1,617	87,741±1,607
	Floresta Aleatória	87,104±2,205	87,178±2,192
	SVM Linear	90,388±1,852	90,400±1,845
	SVM RBF	91,725±1,695	91,756±1,716
VGG16	Bayes	77,919±1,296	78,048±1,222
	MLP	88,297±1,325	88,297±1,337
	k-NN	86,719±1,216	86,700±1,234
	Floresta Aleatória	87,316±1,485	87,300±1,485
	SVM Linear	88,397±1,150	88,434±1,154
	SVM RBF	90,796±1,135	90,820±1,145
VGG19	Bayes	75,623±1,521	75,744±1,540
	MLP	87,107±1,198	87,131±1,195
	k-NN	85,483±1,601	85,441±1,598
	Floresta Aleatória	87,001±0,833	87,033±0,807
	SVM Linear	88,200±1,065	88,260±1,074
	SVM RBF	89,303±1,049	89,328±1,056
ResNet50	Bayes	58,950±2,973	57,864±3,153
	MLP	78,138±1,847	78,131±1,913
	k-NN	70,076±2,084	70,031±2,084
	Floresta Aleatória	73,671±2,528	73,242±2,518
	SVM Linear	78,898±1,736	78,949±1,714
	SVM RBF	79,580±2,172	79,560±2,258
InceptionV3	Bayes	77,100±1,133	77,353±1,097
	MLP	88,243±1,751	88,244±1,755
	k-NN	83,022±1,657	83,275±1,609
	Floresta Aleatória	84,161±1,486	84,173±1,523
	SVM Linear	88,042±1,577	88,077±1,570
	SVM RBF	88,462±1,494	88,485±1,517
ResNet152	Bayes	58,916±1,856	57,803±1,974
	MLP	77,161±1,592	77,123±1,582
	k-NN	70,219±1,506	70,322±1,408
	Floresta Aleatória	74,818±1,308	74,368±1,222
	SVM Linear	78,040±1,376	77,991±1,414
	SVM RBF	80,004±1,307	79,960±1,318

Fonte: Elaborado pelo autor.

desvios padrão parecem ser grandes apenas devido aos limites do eixo y.

Analisando as matrizes de confusão na Figura 9 para a DenseNet169 - SVM (RBF)

Tabela 5 – (Cont.) Acurácia e *F1-Score* obtidos por cada combinação de extrator e classificador de características. Seus respectivos desvios padrão acompanham todas as métricas.

Extrator de características	Classificador	Acurácia	F1-Score
ResNet101	Bayes	60,496±1,128	60,380±1,180
	MLP	77,453±1,436	77,444±1,372
	k-NN	70,516±1,445	70,465±1,427
	Floresta Aleatória	73,833±1,505	73,572±1,460
	SVM Linear	78,373±1,800	78,408±1,784
	SVM RBF	79,275±1,728	79,297±1,696
ResNet152V2	Bayes	77,463±1,438	77,852±1,525
	MLP	88,961±1,673	88,966±1,695
	k-NN	79,556±1,539	79,653±1,563
	Floresta Aleatória	85,322±1,633	85,341±1,657
	SVM Linear	88,664±1,526	88,703±1,511
	SVM RBF	88,981±1,527	88,994±1,511
Xception	Bayes	78,482±1,526	78,503±1,564
	MLP	91,082±1,035	91,080±1,032
	k-NN	85,782±1,782	86,071±1,680
	Floresta Aleatória	87,941±1,227	87,943±1,214
	SVM Linear	90,420±1,283	90,441±1,258
	SVM RBF	91,441±1,199	91,462±1,182
NASNetMobile	Bayes	78,063±2,349	78,187±2,276
	MLP	87,965±1,738	87,950±1,742
	k-NN	82,046±1,809	82,264±1,731
	Floresta Aleatória	85,500±1,910	85,505±1,874
	SVM Linear	88,264±1,926	88,315±1,882
	SVM RBF	89,263±1,704	89,293±1,675

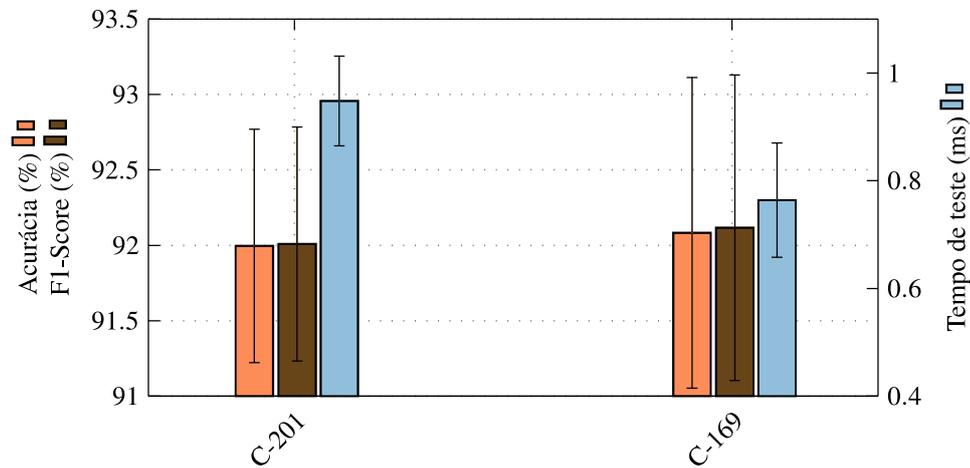
Fonte: Elaborado pelo autor.

e na Figura 10 para DenseNet201 - SVM (RBF), nota-se que essas combinações distinguem claramente as classes 0, 3, 4, 5, 6 e 7, que representam epitélio tumoral, células imunes, detritos, glândula mucosa, tecido adiposo e fundo, respectivamente. A confusão ocorre principalmente nas classes 1 e 2, que são as classes de estroma simples e estroma complexo.

Essa confusão ocorre porque o estroma simples (classe 1) e o estroma complexo (classe 2) podem se sobrepor. Além disso, o estroma complexo apresenta tendência a se agrupar ao redor do epitélio tumoral (classe 3). Portanto, a acurácia e o *F1-Score* para a combinação DenseNet201 - SVM (RBF) são ligeiramente inferiores quando comparadas com essas mesmas métricas para a combinação DenseNet169 com SVM (RBF). Então, a combinação DenseNet169 e SVM (RBF) se destaca pela capacidade de classificar melhor as classes estroma simples (classe 1) e estroma complexo (classe 2).

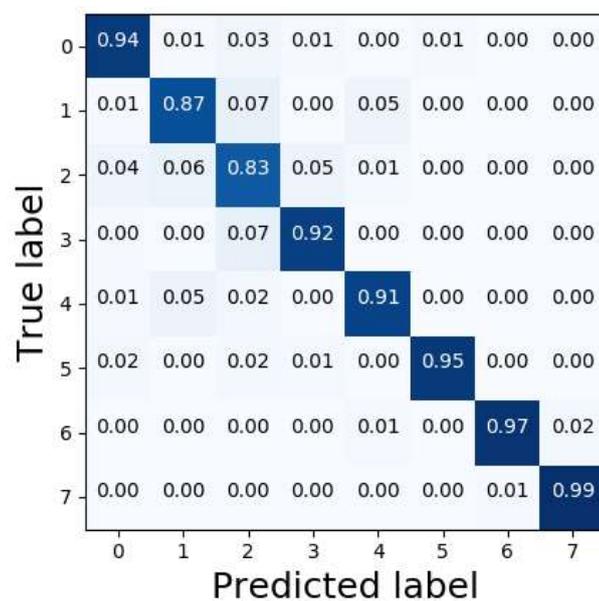
A Figura 11 apresenta as características extraídas utilizando a DenseNet169 usando

Figura 8 – Acurácia, F1-Score e tempo de teste para as melhores combinações de extrator de atributos com classificador. C-201: DenseNet201 + SVM (RBF), C-169: DenseNet169 + SVM (RBF).



Fonte: Elaborado pelo autor.

Figura 9 – Matriz de confusão de DenseNet169 combinada com SVM RBF. *True label* - Classe verdadeira, *Predicted label* - Classe Predita. Classe 0 - epitélio tumoral, Classe 1 - estroma simples, Classe 2 - estroma complexo, Classe 3 - células imunes, Classe 4 - detritos e muco, Classe 5- glândula mucosa, Classe 6 - tecido adiposo e Classe 7 - fundo.



Fonte: Elaborado pelo autor.

Tabela 6 – Tempos de treino e teste obtidos por cada combinação de extrator e classificador de características. Seus respectivos desvios padrão acompanham todas as métricas.

Extrator de características	Classificador	Tempo de Treinamento (s)	Tempo de Teste (ms)
ResNet101V2	Bayes	1,256104±0,667072	0,005079±0,002390
	MLP	4471,702107±238,808700	0,108807±0,069536
	k-NN	2,757085±1,173433	2,037132±0,045359
	Floresta Aleatória	737,463577±37,933863	0,849688±0,015398
	SVM Linear	153,309223±47,390752	0,943807±0,278556
	SVM RBF	376,686515±88,095863	1,548771±0,324230
DenseNet121	Bayes	0,492448±0,290841	0,004071±0,001845
	MLP	1887,720808±61,145177	0,013205±0,006193
	k-NN	1,866221±1,475511	1,724742±0,061296
	Floresta Aleatória	474,442361±11,914351	0,450649±0,009261
	SVM Linear	125,490908±19,927303	0,772196±0,134570
	SVM RBF	208,406669±42,927784	0,949253±0,177107
DenseNet201	Bayes	2,904449±1,722331	0,006207±0,003063
	MLP	557,256617±88,912056	0,004126±0,002062
	k-NN	0,958944±0,415036	1,788717±0,098374
	Floresta Aleatória	408,918562±13,394929	0,231247±0,009132
	SVM Linear	129,101491±33,952520	0,717847±0,182264
	SVM RBF	265,562540±23,886190	0,948397±0,083268
MobileNetV2	Bayes	0,500899±0,116833	0,003779±0,001563
	MLP	1187,066163±109,525459	0,045051±0,022066
	k-NN	1,391544±0,944361	1,546576±0,047736
	Floresta Aleatória	602,477331±24,422236	0,328559±0,020104
	SVM Linear	98,248825±6,716835	0,546791±0,023185
	SVM RBF	239,725331±40,713824	0,780648±0,066941
ResNet50V2	Bayes	0,651892±0,329824	0,004223±0,002002
	MLP	1581,108403±128,437121	0,024782±0,010695
	k-NN	1,408597±0,260703	1,638870±0,042978
	Floresta Aleatória	247,085370±6,520427	0,179283±0,006390
	SVM Linear	132,334184±29,790369	0,817595±0,090858
	SVM RBF	259,309113±22,694628	1,103358±0,226573
NASNetLarge	Bayes	1,394583±0,524192	0,006003±0,002115
	MLP	2323,706979±282,362831	0,048789±0,018099
	k-NN	2,277660±0,429664	3,430507±0,181021
	Floresta Aleatória	806,373208±27,908470	0,385989±0,030136
	SVM Linear	284,737513±31,351743	1,583620±0,162267
	SVM RBF	412,473736±6,816471	2,417776±0,577686
InceptionResNetV2	Bayes	1,273082±0,467151	0,005708±0,002597
	MLP	676,657151±97,079157	0,004195±0,002378
	k-NN	1,511736±1,447230	1,393645±0,044317
	Floresta Aleatória	748,135841±30,579288	0,269957±0,011832
	SVM Linear	106,486707±22,140289	0,593405±0,051671
	SVM RBF	119,673637±18,855987	0,635505±0,054499
DenseNet169	Bayes	0,858342±0,541357	0,006008±0,001624
	MLP	1680,285246±109,458078	0,014161±0,006900
	k-NN	2,201647±2,210836	1,937445±0,083650
	Floresta Aleatória	1432,010970±61,574034	0,788451±0,022768
	SVM Linear	213,596518±27,513107	1,027755±0,014714
	SVM RBF	179,115986±40,839740	0,764738±0,106099

Fonte: Elaborado pelo autor.

Tabela 6 – (Cont.) Tempos de treino e teste obtidos por cada combinação de extrator e classificador de características. Seus respectivos desvios padrão acompanham todas as métricas.

Extrator de características	Classificador	Tempo de Treinamento (s)	Tempo de Teste (ms)
MobileNet	Bayes	0,546731±0,500385	0,002953±0,001838
	MLP	546,496072±67,986574	0,015247±0,003865
	k-NN	2,142582±1,968038	1,023620±0,139765
	Floresta Aleatória	438,926362±12,416899	0,449182±0,023301
	SVM Linear	70,660073±15,417306	0,464747±0,118563
	SVM RBF	130,809587±12,778148	0,614404±0,183073
VGG16	Bayes	0,144882±0,052616	0,004564±0,001830
	MLP	1729,524620±107,380049	0,009009±0,004564
	k-NN	0,350058±0,155364	0,529276±0,027895
	Floresta Aleatória	877,775962±35,005706	0,794564±0,017710
	SVM Linear	47,368195±15,644190	0,236372±0,060112
	SVM RBF	64,628976±2,776642	0,379487±0,029866
VGG19	Bayes	0,212815±0,075386	0,002374±0,001683
	MLP	1200,641323±78,660175	0,009521±0,005240
	k-NN	0,269711±0,148897	0,498642±0,091794
	Floresta Aleatória	110,925332±3,459166	0,393141±0,040268
	SVM Linear	26,258812±4,735973	0,187476±0,036375
	SVM RBF	41,094585±14,761574	0,221090±0,056550
ResNet50	Bayes	1,366463±0,906024	0,007493±0,002562
	MLP	2548,602765±235,377757	0,017675±0,009857
	k-NN	2,255636±0,577269	1,614571±0,131339
	Floresta Aleatória	597,865272±28,408460	0,551806±0,018161
	SVM Linear	230,778445±69,345011	1,246999±0,371225
	SVM RBF	316,183878±104,676711	1,576447±0,442080
InceptionV3	Bayes	0,924846±0,322379	0,004046±0,001171
	MLP	3056,391673±294,808432	0,046383±0,026891
	k-NN	2,667241±2,347622	2,052711±0,056952
	Floresta Aleatória	1687,597282±69,116237	0,487001±0,045884
	SVM Linear	176,472038±36,993404	1,127923±0,094545
	SVM RBF	353,725177±57,100580	1,387665±0,213145
ResNet152	Bayes	0,636714±0,286117	0,004254±0,001691
	MLP	6981,808846±226,768505	0,040989±0,039438
	k-NN	1,946031±0,143307	1,869638±0,040302
	Floresta Aleatória	355,110181±15,749700	0,355637±0,010677
	SVM Linear	265,155542±59,374448	1,236719±0,234938
	SVM RBF	280,955045±28,556761	1,147347±0,102684
ResNet101	Bayes	1,146787±0,419369	0,005764±0,001700
	MLP	4074,093161±155,083086	0,023899±0,012961
	k-NN	2,597779±0,108729	2,208008±0,049697
	Floresta Aleatória	96,973143±5,019693	0,323434±0,007636
	SVM Linear	257,035909±64,325011	1,342147±0,215593
	SVM RBF	358,611006±7,339549	1,104626±0,021773

Fonte: Elaborado pelo autor.

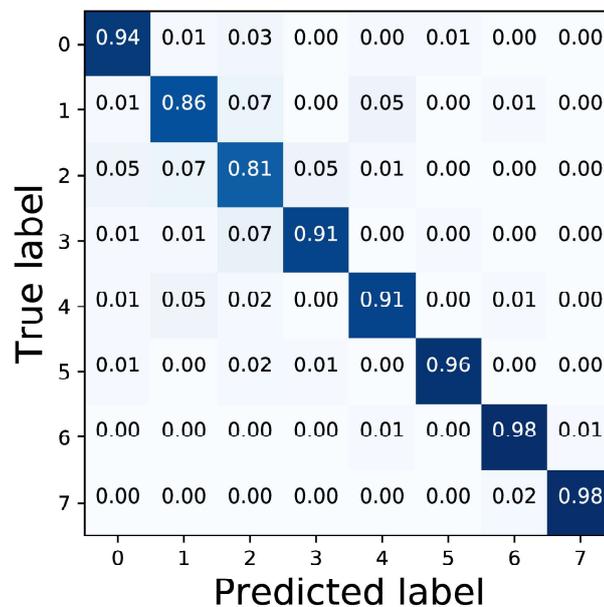
o t-SNE (MAATEN; HINTON, 2008). É possível de observar que as classes não são linearmente separáveis, porém há alguns agrupamentos claramente distinguíveis. Entre as classes que possuem os agrupamentos distinguíveis são as classes 5, 6 e 7, no qual representam glândula

Tabela 6 – (Cont.) Tempos de treino e teste obtidos por cada combinação de extrator e classificador de características. Seus respectivos desvios padrão acompanham todas as métricas.

Extrator de características	Classificador	Tempo de Treinamento (s)	Tempo de Teste (ms)
ResNet152V2	Bayes	0,729931±0,278260	0,003357±0,000932
	MLP	1931,446265±251,967927	0,042560±0,020959
	k-NN	3,373562±0,973163	2,484569±0,247448
	Floresta Aleatória	1122,204345±49,762525	0,815479±0,024736
	SVM Linear	209,961904±47,978272	1,218995±0,253129
	SVM RBF	297,001313±85,921826	1,983845±0,070923
Xception	Bayes	1,118301±0,717466	0,005578±0,003523
	MLP	3357,911573±259,828742	0,106830±0,043945
	k-NN	2,582450±1,055500	2,001393±0,083567
	Floresta Aleatória	1127,323146±31,820432	0,609433±0,022870
	SVM Linear	138,670975±2,668562	0,853885±0,084326
	SVM RBF	167,276784±33,990876	1,164898±0,356653
NASNetMobile	Bayes	0,512117±0,345216	0,004085±0,002113
	MLP	1199,904453±158,429208	0,010047±0,005943
	k-NN	0,358513±0,020915	1,058878±0,034788
	Floresta Aleatória	143,013124±4,656400	0,332898±0,018342
	SVM Linear	69,613175±4,740637	0,390025±0,031622
	SVM RBF	210,921591±37,106153	0,576849±0,042696

Fonte: Elaborado pelo autor.

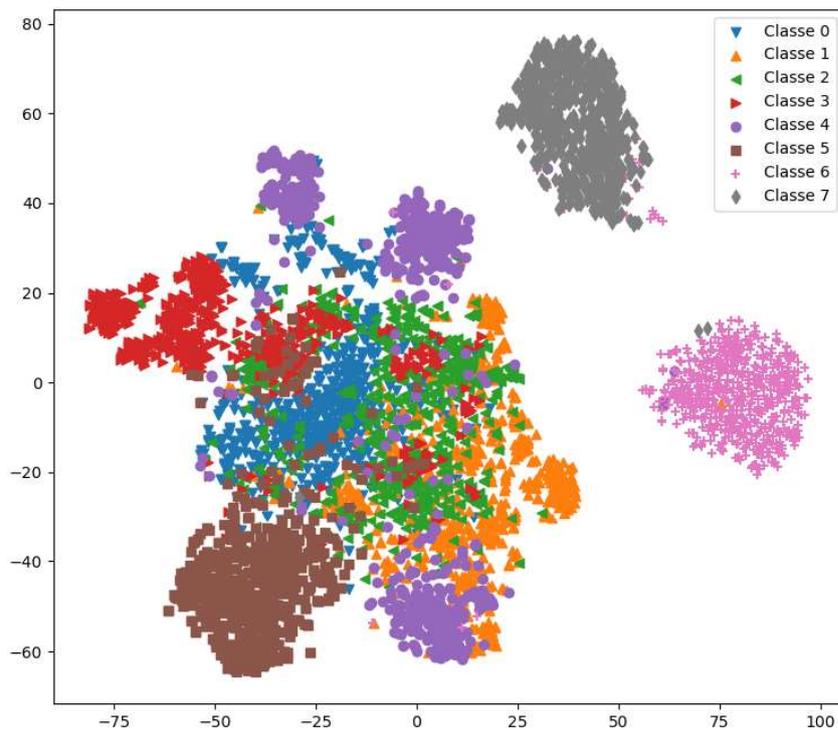
Figura 10 – Matriz de confusão da DenseNet201 combinada com SVM RBF. *True label* - Classe verdadeira, *Predicted label* - Classe Predita. Classe 0 - epitélio tumoral, Classe 1 - estroma simples, Classe 2 - estroma complexo, Classe 3 - células imunes, Classe 4 - detritos e muco, Classe 5- glândula mucosa, Classe 6 - tecido adiposo e Classe 7 - fundo.



Fonte: Elaborado pelo autor.

mucosa, tecido adiposo e fundo, respectivamente. Observando amostras dessas classes, pode-se notar que elas tem características únicas, o que resulta na alta taxa de acerto dessas classes observada na Figura 9. Assim como também é possível verificar a alta sobreposição das classes 2 e 3, e confirmando o desafio de distinguir essas classes.

Figura 11 – Visualização dos atributos extraídos com a DenseNet169 utilizando t-SNE. Classe 0 - epitélio tumoral, Classe 1 - estroma simples, Classe 2 - estroma complexo, Classe 3 - células imunes, Classe 4 - detritos e muco, Classe 5 - glândula mucosa, Classe 6 - tecido adiposo e Classe 7 - fundo.



Fonte: Elaborado pelo autor.

5.2 Comparação com outros métodos no conjunto de dados *Colorectal Cancer Histology*

Os resultados da classificação de imagens histológicas de CRC utilizando CNN como extratores de características através da técnica de *Transfer Learning*, e classificando-as com métodos de aprendizado de máquina são comparados com outros estudos da literatura que utilizam o mesmo conjunto de dados (*Colorectal Cancer Histology*).

A Tabela 7 mostra que a abordagem proposta por Wang *et al.* (2017) atingiu uma acurácia 0,52% maior do que a abordagem proposta neste trabalho. No entanto, eles selecionaram apenas 1000 imagens não sobrepostas. Essa seleção pode ter influenciado o resultado, uma vez

que imagens difíceis de classificar podem ter sido removidas; por exemplo, os autores podem ter removido as imagens mais difíceis do estroma simples e classes de estroma complexo, que podem se sobrepor.

Rachapudi e Devi (2020) não apresentaram a acurácia geral; os autores realizaram uma análise por classe e apresentaram uma TE de 22,7%. Na abordagem proposta por esta dissertação atingiu um TE de 7,9%, demonstrando que teve um desempenho melhor do que a CNN proposta por Rachapudi e Devi (2020).

O método proposto por Kather *et al.* (2016) é o método mais comparável à abordagem proposta. Ambas as abordagens realizaram extração extensiva de atributos, e usaram diferentes técnicas de aprendizado de máquina. No entanto, Kather *et al.* (2016) aplicou extratores de características com base em textura e fez combinações dos conjuntos de características extraídas. Na Tabela 7, pode-se observar que a abordagem proposta obteve uma acurácia 4,68% maior que a abordagem proposta por Kather *et al.* (2016). Portanto, embora Kather *et al.* (2016) tenha combinados os conjuntos de características extraídas, as CNN podem extrair recursos mais relevantes das imagens.

Tabela 7 – Comparação dos resultados obtidos pelo método proposto e outras abordagens.

Autores	Quantidade de imagens	Abordagem	Método	Resultados
Método proposto	5000	Extração de características usando CNNs por meio de <i>Transfer Learning</i> e, em seguida, classificação com métodos de aprendizagem de máquina	DenseNet169 + SVM (RBF)	Acurácia: 92,08% F1-Score: 92,12% TE: 7,9%
Wang <i>et al.</i> (2017)	1000	Decompõe os componentes H&E e, em seguida, aplicado a rede baseada na BCNN para classificar as imagens decompostas	Decomposição + BCNN	Acurácia: 92,6%
Rachapudi e Devi (2020)	5000	Uma nova arquitetura de CNN	CNN com cinco blocos de camadas	TE: 22,7%
Kather <i>et al.</i> (2016)	5000	Extração de características combinando métodos tradicionais de extração de atributos e, em seguida, classificação com métodos de aprendizado de máquina	Combinação de técnicas de extração de características + SVM (RBF)	Acurácia: 87,4% TE: 12,6%

Fonte: Elaborado pelo autor.

6 CONCLUSÃO E TRABALHOS FUTUROS

Esta dissertação apresenta um método para classificar imagens de tecidos da histologia do CRC em oito classes. A abordagem é dividida em duas etapas: i) extração de características usando CNN por meio da técnica de *Transfer Learning* e ii) classificação usando métodos de aprendizado de máquina.

Foi realizado uma ampla combinação de extratores de características e classificadores, resultando em 108 experimentos, usando o método *Transfer Learning*. Os resultados mostram que CNN, combinadas com a técnica de *Transfer Learning* e métodos de aprendizagem de máquina, podem ser usados como extratores de atributos para este problema. O extrator de características DenseNet169 combinado com SVM (RBF) produziu o melhor desempenho, obtendo 92,08% de Acurácia, 92,117% de F1-Score e 0,76 ms no tempo de teste, o que significa baixo custo computacional.

Portanto, este método se mostrou eficiente em classificar as imagens histológicas do CRC com alta precisão e rapidez. Assim, pode ser uma ferramenta útil para auxiliar médicos e técnicos a identificar o tipo correto de tecido neste exame. Por esse motivo, as amostras cancerosas têm menos probabilidade de passar despercebidas e os pacientes receberão tratamentos adequados e precoces com mais frequência.

Para trabalhos futuros, pretende-se expandir a aplicação da classificação automática de tecidos em exames histológicos. Inicialmente, pretende-se testar o método proposto em diferentes conjuntos de dados histológicos. Em seguida, planeja-se combinar diferentes CNN e até métodos tradicionais de extração de atributos, com o objetivo de melhorar a solução. Além disso, pretende-se criar um sistema de Internet das Coisas para que essa abordagem seja acessível, principalmente, nos países em desenvolvimento, onde encontra-se as maiores taxas de incidência e mortalidade do CRC. Por fim, pretende-se analisar a viabilidade do sistema com o sistema PACS.

REFERÊNCIAS

- ABIR, F.; ALVA, S.; LONGO, W. E.; AUDISO, R.; VIRGO, K. S.; JOHNSON, F. E. The postoperative surveillance of patients with colon cancer and rectal cancer. **The American journal of surgery**, Elsevier, v. 192, n. 1, p. 100–108, 2006.
- ALHINDI, T. J.; KALRA, S.; NG, K. H.; AFRIN, A.; TIZHOOSH, H. R. Comparing lbp, hog and deep features for classification of histopathology images. In: IEEE. **2018 international joint conference on neural networks (IJCNN)**. [S. l.], 2018. p. 1–7.
- ARAÚJO, T.; ARESTA, G.; CASTRO, E.; ROUCO, J.; AGUIAR, P.; ELOY, C.; POLÓNIA, A.; CAMPILHO, A. Classification of breast cancer histology images using convolutional neural networks. **PLOS ONE**, Public Library of Science (PLoS), v. 12, n. 6, p. e0177544, jun. 2017. Disponível em: <https://doi.org/10.1371/journal.pone.0177544>.
- ARNOLD, M.; SIERRA, M. S.; LAVERSANNE, M.; SOERJOMATARAM, I.; JEMAL, A.; BRAY, F. Global patterns and trends in colorectal cancer incidence and mortality. **Gut**, BMJ Publishing Group, v. 66, n. 4, p. 683–691, 2017.
- BAYRAMOGLU, N.; HEIKKILÄ, J. Transfer learning for cell nuclei classification in histopathology images. In: SPRINGER. **European Conference on Computer Vision**. [S. l.], 2016. p. 532–539.
- BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. **IEEE transactions on neural networks**, IEEE, v. 5, n. 2, p. 157–166, 1994.
- BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001.
- CHEN, W.; ZHANG, Z. Hand gesture recognition using semg signals based on support vector machine. In: IEEE. **2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)**. [S. l.], 2019. p. 230–234.
- CHOI, J. Y.; YOO, T. K.; SEO, J. G.; KWAK, J.; UM, T. T.; RIM, T. H. Multi-categorical deep learning neural network to classify retinal images: a pilot study employing small database. **PloS one**, Public Library of Science, v. 12, n. 11, 2017.
- CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S. l.: s. n.], 2017. p. 1251–1258.
- COUDRAY, N.; OCAMPO, P. S.; SAKELLAROPOULOS, T.; NARULA, N.; SNUDERL, M.; FENYÖ, D.; MOREIRA, A. L.; RAZAVIAN, N.; TSIRIGOS, A. Classification and mutation prediction from non-small cell lung cancer histopathology images using deep learning. **Nature medicine**, Nature Publishing Group, v. 24, n. 10, p. 1559–1567, 2018.
- DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In: IEEE. **2009 IEEE conference on computer vision and pattern recognition**. [S. l.], 2009. p. 248–255.
- DINO, H. I.; ABDULRAZZAQ, M. B. Facial expression classification based on svm, knn and mlp classifiers. In: IEEE. **2019 International Conference on Advanced Science and Engineering (ICOASE)**. [S. l.], 2019. p. 70–75.

FERLAY, J.; SOERJOMATARAM, I.; DIKSHIT, R.; ESER, S.; MATHERS, C.; REBELO, M.; PARKIN, D. M.; FORMAN, D.; BRAY, F. Cancer incidence and mortality worldwide: sources, methods and major patterns in globocan 2012. **International journal of cancer**, Wiley Online Library, v. 136, n. 5, p. E359–E386, 2015.

FUKUNAGA, K.; NARENDRA, P. M. A branch and bound algorithm for computing k-nearest neighbors. **IEEE Transactions on Computers**, C-24, n. 7, p. 750–753, July 1975. ISSN 0018-9340.

GAO, S.; ZHOU, M.; WANG, Y.; CHENG, J.; YACHI, H.; WANG, J. Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction. **IEEE transactions on neural networks and learning systems**, IEEE, v. 30, n. 2, p. 601–614, 2018.

GECER, B.; AKSOY, S.; MERCAN, E.; SHAPIRO, L. G.; WEAVER, D. L.; ELMORE, J. G. Detection and classification of cancer in whole slide breast histopathology images using deep convolutional networks. **Pattern recognition**, Elsevier, v. 84, p. 345–356, 2018.

GIRARD, L.; RODRIGUEZ-CANALES, J.; BEHRENS, C.; THOMPSON, D. M.; BOTROS, I. W.; TANG, H.; XIE, Y.; REKHTMAN, N.; TRAVIS, W. D.; WISTUBA, I. I. *et al.* An expression signature as an aid to the histologic classification of non–small cell lung cancer. **Clinical Cancer Research**, AACR, v. 22, n. 19, p. 4880–4889, 2016.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge, Massachusetts: MIT Press, 2016.

GU, J.; WANG, Z.; KUEN, J.; MA, L.; SHAHROUDY, A.; SHUAI, B.; LIU, T.; WANG, X.; WANG, G.; CAI, J. *et al.* Recent advances in convolutional neural networks. **Pattern Recognition**, Elsevier, v. 77, p. 354–377, 2018.

HAYKIN, S. Neural networks: principles and practice. **Bookman**, 2001.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S. l.: s. n.], 2016. p. 770–778.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Identity mappings in deep residual networks. In: SPRINGER. **European conference on computer vision**. [S. l.], 2016. p. 630–645.

HEBBAR, A.; KUMAR, M.; SANJAY, H. Drap: Decision tree and random forest based classification model to predict diabetes. In: IEEE. **2019 1st International Conference on Advances in Information Technology (ICAIT)**. [S. l.], 2019. p. 271–276.

HINTON, G.; VINYALS, O.; DEAN, J. Distilling the knowledge in a neural network. **arXiv preprint arXiv:1503.02531**, 2015.

HO, T. K. Random decision forests. In: IEEE. **Proceedings of 3rd international conference on document analysis and recognition**. [S. l.], 1995. v. 1, p. 278–282.

HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, 2017.

- HUANG, G.; LIU, Z.; MAATEN, L. V. D.; WEINBERGER, K. Q. Densely connected convolutional networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S. l.: s. n.], 2017. p. 4700–4708.
- HUH, M.; AGRAWAL, P.; EFROS, A. A. What makes imagenet good for transfer learning? **arXiv preprint arXiv:1608.08614**, 2016.
- HUSSEIN, S.; CAO, K.; SONG, Q.; BAGCI, U. Risk stratification of lung nodules using 3d CNN-based multi-task learning. In: **Lecture Notes in Computer Science**. [S. l.]: Springer International Publishing, 2017. p. 249–260.
- IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: PMLR. **International conference on machine learning**. [S. l.], 2015. p. 448–456.
- JIN, J.; DUNDAR, A.; CULURCIELLO, E. Flattened convolutional neural networks for feedforward acceleration. **arXiv preprint arXiv:1412.5474**, 2014.
- JR, C. M. D.; SILVA, S. P. P. da; NÓBREGA, R. V. M. da; BARROS, A. C. da S.; FILHO, P. P. R.; ALBUQUERQUE, V. H. C. de. Deep learning IoT system for online stroke detection in skull computed tomography images. **Computer Networks**, Elsevier BV, v. 152, p. 25–39, abr. 2019. Disponível em: <https://doi.org/10.1016/j.comnet.2019.01.019>.
- KATHER, J.; WEIS, C.-A.; BIANCONI, F.; MELCHERS, S.; SCHAD, L.; GAISER, T.; MARX, A.; ZÖLLNER, F. Multi-class texture analysis in colorectal cancer histology. **Scientific Reports**, v. 6, p. 27988, 06 2016.
- KATHER, J. N.; KRISAM, J.; CHAROENTONG, P.; LUEDDE, T.; HERPEL, E.; WEIS, C.-A.; GAISER, T.; MARX, A.; VALOUS, N. A.; FERBER, D. *et al.* Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. **PLoS medicine**, Public Library of Science, v. 16, n. 1, 2019.
- KATHER, J. N.; ZÖLLNER, F. G.; BIANCONI, F.; MELCHERS, S. M.; SCHAD, L. R.; GAISER, T.; MARX, A.; WEIS, C.-A. **Collection Of Textures In Colorectal Cancer Histology**. Zenodo, 2016. Disponível em: <https://zenodo.org/record/53169>.
- KERMANY, D. S.; GOLDBAUM, M.; CAI, W.; VALENTIM, C. C.; LIANG, H.; BAXTER, S. L.; MCKEOWN, A.; YANG, G.; WU, X.; YAN, F.; DONG, J.; PRASADHA, M. K.; PEI, J.; TING, M. Y.; ZHU, J.; LI, C.; HEWETT, S.; DONG, J.; ZIYAR, I.; SHI, A.; ZHANG, R.; ZHENG, L.; HOU, R.; SHI, W.; FU, X.; DUAN, Y.; HUU, V. A.; WEN, C.; ZHANG, E. D.; ZHANG, C. L.; LI, O.; WANG, X.; SINGER, M. A.; SUN, X.; XU, J.; TAFRESHI, A.; LEWIS, M. A.; XIA, H.; ZHANG, K. Identifying medical diagnoses and treatable diseases by image-based deep learning. **Cell**, Elsevier BV, v. 172, n. 5, p. 1122–1131.e9, fev. 2018. Disponível em: <https://doi.org/10.1016/j.cell.2018.02.010>.
- KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.
- KRIZHEVSKY, A.; HINTON, G. *et al.* Learning multiple layers of features from tiny images. Citeseer, 2009.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. Imagenet classification with deep convolutional neural networks. In: . [S. l.: s. n.], 2012. v. 2, p. 1097–1105.

- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015.
- LECUN, Y.; BOSER, B.; DENKER, J.; HENDERSON, D.; HOWARD, R.; HUBBARD, W.; JACKEL, L. Handwritten digit recognition with a back-propagation network. **Advances in neural information processing systems**, v. 2, 1989.
- LECUN, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998.
- LIN, T.-Y.; MAIRE, M.; BELONGIE, S.; HAYS, J.; PERONA, P.; RAMANAN, D.; DOLLÁR, P.; ZITNICK, C. L. Microsoft coco: Common objects in context. In: SPRINGER. **European conference on computer vision**. [S. l.], 2014. p. 740–755.
- LIN, Y.; JEON, Y. Random forests and adaptive nearest neighbors. **Journal of the American Statistical Association**, Taylor & Francis, v. 101, n. 474, p. 578–590, 2006.
- LOCONSOLE, C.; CASCARANO, G. D.; LATTARULO, A.; BRUNETTI, A.; TROTTA, G. F.; BUONGIORNO, D.; BORTONE, I.; FEUDIS, I. D.; LOSAVIO, G.; BEVILACQUA, V. *et al.* A comparison between ANN and SVM classifiers for parkinson’s disease by using a model-free computer-assisted handwriting analysis based on biometric signals. In: IEEE. **2018 International Joint Conference on Neural Networks (IJCNN)**. [S. l.], 2018. p. 1–8.
- LOH, B. C. S.; THEN, P. H. H. Deep learning for cardiac computer-aided diagnosis: benefits, issues & solutions. **mHealth**, AME Publishing Company, v. 3, p. 45–45, out. 2017. Disponível em: <https://doi.org/10.21037/mhealth.2017.09.01>.
- MAATEN, L. v. d.; HINTON, G. Visualizing data using t-sne. **Journal of Machine Learning Research**, v. 9, n. Nov, p. 2579–2605, 2008.
- NÓBREGA, R. V. M. da; FILHO, P. P. R.; RODRIGUES, M. B.; SILVA, S. P. P. da; JÚNIOR, C. M. J. M. D.; ALBUQUERQUE, V. H. C. de. Lung nodule malignancy classification in chest computed tomography images using transfer learning and convolutional neural networks. **Neural Computing and Applications**, Springer Science and Business Media LLC, nov. 2018. Disponível em: <https://doi.org/10.1007/s00521-018-3895-1>.
- ORENSTEIN, E. C.; BEIJBOM, O. Transfer learning and deep feature extraction for planktonic image data sets. In: IEEE. **2017 IEEE Winter Conference on Applications of Computer Vision (WACV)**. [S. l.], 2017. p. 1082–1088.
- PAN, S. J.; YANG, Q. A survey on transfer learning. **IEEE Transactions on knowledge and data engineering**, IEEE, v. 22, n. 10, p. 1345–1359, 2009.
- PAUL, R.; HAWKINS, S. H.; BALAGURUNATHAN, Y.; SCHABATH, M. B.; GILLIES, R. J.; HALL, L. O.; GOLDFOF, D. B. Deep feature transfer learning in combination with traditional features predicts survival among patients with lung adenocarcinoma. **Tomography**, Grapho Publications, v. 2, n. 4, p. 388, 2016.
- RACHAPUDI, V.; DEVI, G. L. Improved convolutional neural network based histopathological image classification. **Evolutionary Intelligence**, Springer, p. 1–7, 2020.
- RODRIGUES, J. d. C.; FILHO, P. P. R.; JR, E. P.; KUMAR, A.; ALBUQUERQUE, V. H. C. de. Classification of eeg signals to detect alcoholism using machine learning techniques. **Pattern Recognition Letters**, Elsevier, v. 125, p. 140–149, 2019.

- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958.
- RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATY, A.; KHOSLA, A.; BERNSTEIN, M. *et al.* Imagenet large scale visual recognition challenge. **International journal of computer vision**, Springer, v. 115, n. 3, p. 211–252, 2015.
- SANDLER, M.; HOWARD, A.; ZHU, M.; ZHMOGINOV, A.; CHEN, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S. l.: s. n.], 2018. p. 4510–4520.
- SANTOS, M. A.; MUNOZ, R.; OLIVARES, R.; FILHO, P. P. R.; SER, J. D.; ALBUQUERQUE, V. H. C. de. Online heart monitoring systems on the internet of health things environments: A survey, a reference model and an outlook. **Information Fusion**, Elsevier, v. 53, p. 222–239, 2020.
- SARMENTO, R. M.; VASCONCELOS, F. F.; FILHO, P. P. R.; ALBUQUERQUE, V. H. C. de. An IoT platform for the analysis of brain CT images based on parzen analysis. **Future Generation Computer Systems**, Elsevier BV, v. 105, p. 135–147, abr. 2020.
- SARMENTO, R. M.; VASCONCELOS, F. F. X.; FILHO, P. P. R.; WU, W.; ALBUQUERQUE, V. H. C. D. Automatic neuroimage processing and analysis in stroke—a systematic review. **IEEE reviews in biomedical engineering**, IEEE, v. 13, p. 130–155, 2019.
- SHABAN, M.; AWAN, R.; FRAZ, M. M.; AZAM, A.; TSANG, Y.-W.; SNEAD, D.; RAJPOOT, N. M. Context-aware convolutional neural network for grading of colorectal cancer histology images. **IEEE Transactions on Medical Imaging**, Institute of Electrical and Electronics Engineers (IEEE), p. 1–1, 2020. Disponível em: <https://doi.org/10.1109/tmi.2020.2971006>.
- SIEGEL, R. L.; FEDEWA, S. A.; ANDERSON, W. F.; MILLER, K. D.; MA, J.; ROSENBERG, P. S.; JEMAL, A. Colorectal cancer incidence patterns in the united states, 1974–2013. **JNCI: Journal of the National Cancer Institute**, Oxford University Press, v. 109, n. 8, 2017.
- SIEGEL, R. L.; MILLER, K. D.; FEDEWA, S. A.; AHNEN, D. J.; MEESTER, R. G.; BARZI, A.; JEMAL, A. Colorectal cancer statistics, 2017. **CA: a cancer journal for clinicians**, Wiley Online Library, v. 67, n. 3, p. 177–193, 2017.
- SIEGEL, R. L.; MILLER, K. D.; SAUER, A. G.; FEDEWA, S. A.; BUTTERLY, L. F.; ANDERSON, J. C.; CERCEK, A.; SMITH, R. A.; JEMAL, A. Colorectal cancer statistics, 2020. **CA: A Cancer Journal for Clinicians**, Wiley, mar. 2020. Disponível em: <https://doi.org/10.3322/caac.21601>.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.
- SINGH, G.; KUMAR, B.; GAUR, L.; TYAGI, A. Comparison between multinomial and bernoulli naïve bayes for text classification. In: IEEE. **2019 International Conference on Automation, Computational and Technology Management (ICACTM)**. [S. l.], 2019. p. 593–596.
- SLAOUI, M.; FIETTE, L. Histopathology procedures: From tissue sampling to histopathological evaluation. In: **Methods in Molecular Biology**. [S. l.]: Humana Press, 2010. p. 69–82.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. **The journal of machine learning research**, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.

STONE, M. Cross-validatory choice and assessment of statistical predictions. **Journal of the royal statistical society: Series B (Methodological)**, Wiley Online Library, v. 36, n. 2, p. 111–133, 1974.

SZEGEDY, C.; IOFFE, S.; VANHOUCKE, V.; ALEMI, A. A. Inception-v4, inception-resnet and the impact of residual connections on learning. In: **Thirty-first AAAI conference on artificial intelligence**. [S. l.: s. n.], 2017.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S. l.: s. n.], 2015. p. 1–9.

TANG, H.; SCAIFE, A. M.; LEAHY, J. Transfer learning for radio galaxy classification. **arXiv preprint arXiv:1903.11921**, 2019.

THEODORIDIS, S.; PIKRAKIS, A.; KOUTROUMBAS, K.; CAVOURAS, D. **Introduction to pattern recognition: a matlab approach**. [S. l.]: Academic Press, 2010.

WAN, S.; LIANG, Y.; ZHANG, Y.; GUIZANI, M. Deep multi-layer perceptron classifier for behavior analysis to estimate parkinson's disease severity using smartphones. **IEEE Access**, IEEE, v. 6, p. 36825–36833, 2018.

WANG, C.; SHI, J.; ZHANG, Q.; YING, S. Histopathological image classification with bilinear convolutional neural networks. In: IEEE. **2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)**. [S. l.], 2017. p. 4050–4053.

WANG, P.; HU, X.; LI, Y.; LIU, Q.; ZHU, X. Automatic cell nuclei segmentation and classification of breast cancer histopathology images. **Signal Processing**, Elsevier, v. 122, p. 1–13, 2016.

WANG, X. Ladle furnace temperature prediction model based on large-scale data with random forest. **IEEE/CAA Journal of Automatica Sinica**, IEEE, v. 4, n. 4, p. 770–774, 2016.

WEISS, K.; KHOSHGOFTAAR, T. M.; WANG, D. A survey of transfer learning. **Journal of Big data**, SpringerOpen, v. 3, n. 1, p. 9, 2016.

WU, Z.; XU, Q.; LI, J.; FU, C.; XUAN, Q.; XIANG, Y. Passive indoor localization based on csi and naive bayes classification. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, IEEE, v. 48, n. 9, p. 1566–1577, 2017.

XING, W.; BEI, Y. Medical health big data classification based on knn classification algorithm. **IEEE Access**, IEEE, v. 8, p. 28808–28819, 2019.

XU, J.; XIANG, L.; LIU, Q.; GILMORE, H.; WU, J.; TANG, J.; MADABHUSHI, A. Stacked sparse autoencoder (ssae) for nuclei detection on breast cancer histopathology images. **IEEE transactions on medical imaging**, IEEE, v. 35, n. 1, p. 119–130, 2015.

- XU, Y.; JIA, Z.; WANG, L.-B.; AI, Y.; ZHANG, F.; LAI, M.; ERIC, I.; CHANG, C. Large scale tissue histopathology image classification, segmentation, and visualization via deep convolutional activation features. **BMC bioinformatics**, BioMed Central, v. 18, n. 1, p. 281, 2017.
- YU, K.-H.; WANG, F.; BERRY, G. J.; RÉ, C.; ALTMAN, R. B.; SNYDER, M.; KOHANE, I. S. Classifying non-small cell lung cancer types and transcriptomic subtypes using convolutional neural networks. **Journal of the American Medical Informatics Association**, Oxford University Press (OUP), v. 27, n. 5, p. 757–769, maio 2020. Disponível em: <https://doi.org/10.1093/jamia/ocz230>.
- YU, K.-H.; ZHANG, C.; BERRY, G. J.; ALTMAN, R. B.; RÉ, C.; RUBIN, D. L.; SNYDER, M. Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features. **Nature communications**, Nature Publishing Group, v. 7, p. 12474, 2016.
- ZAW, H. T.; MANEERAT, N.; WIN, K. Y. Brain tumor detection based on naïve bayes classification. In: IEEE. **2019 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST)**. [S. l.], 2019. p. 1–4.
- ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: SPRINGER. **European conference on computer vision**. [S. l.], 2014. p. 818–833.
- ZHANG, H.; FU, Z.; SHU, K.-I. Recognizing ping-pong motions using inertial data based on machine learning classification algorithms. **IEEE Access**, IEEE, v. 7, p. 167055–167064, 2019.
- ZHANG, P.; SHU, S.; ZHOU, M. An online fault detection model and strategies based on svm-grid in clouds. **IEEE/CAA Journal of Automatica Sinica**, IEEE, v. 5, n. 2, p. 445–456, 2018.
- ZHANG, S.; LI, X.; ZONG, M.; ZHU, X.; WANG, R. Efficient knn classification with different numbers of nearest neighbors. **IEEE transactions on neural networks and learning systems**, IEEE, v. 29, n. 5, p. 1774–1785, 2017.
- ZHAO, W. Research on the deep learning of the small sample data based on transfer learning. In: . Author(s), 2017. Disponível em: <https://doi.org/10.1063/1.4992835>.
- ZHU, D.; ZHU, H.; LIU, X.; LI, H.; WANG, F.; LI, H. Achieve efficient and privacy-preserving medical primary diagnosis based on knn. In: IEEE. **2018 27th International Conference on Computer Communication and Networks (ICCCN)**. [S. l.], 2018. p. 1–9.
- ZOPH, B.; LE, Q. V. Neural architecture search with reinforcement learning. **arXiv preprint arXiv:1611.01578**, 2016.