

UNIVERSIDADE DO VALE DO RIO DOS SINOS — UNISINOS

PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA
NÍVEL MESTRADO

FABIANO GODOIS TEIXEIRA

**DETECÇÃO E CLASSIFICAÇÃO DA ALTA EXPRESSÃO DA PROTEÍNA
E-CADERINA NO CARCINOMA DA MAMA ATRAVÉS DE UMA REDE NEURAL
ARTIFICIAL**

São Leopoldo
2020

Fabiano Godois Teixeira

**DETECÇÃO E CLASSIFICAÇÃO DA ALTA EXPRESSÃO DA PROTEÍNA
E-CADERINA NO CARCINOMA DA MAMA ATRAVÉS DE UMA REDE NEURAL
ARTIFICIAL**

Proposta de Dissertação apresentada como
requisito parcial para a obtenção do título de
Mestre pelo Programa de Pós-Graduação em
Computação Aplicada da Universidade do Vale
do Rio dos Sinos — UNISINOS

Orientador:
Prof. Dr. Cristiano André da Costa

São Leopoldo
2020

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)

Teixeira, Fabiano Godois

DETECÇÃO E CLASSIFICAÇÃO DA ALTA EXPRESSÃO DA PROTEÍNA E-CADERINA NO CARCINOMA DA MAMA ATRAVÉS DE UMA REDE NEURAL ARTIFICIAL / Fabiano Godois Teixeira — 2020.

90 f.: il.; 30 cm.

Tese (mestrado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, São Leopoldo, 2020.

“Orientador: Prof. Dr. Cristiano André da Costa, ”.

1. Mamas - Câncer. 2. Inteligência artificial. 3. Inovações disruptivas. 4. Redes neurais (computação). I. Título.

CDU 004.8

Bibliotecária responsável: Bruna Rosa de Sant'Anna — CRB 10/2360

(O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Código de Financiamento 001. /"This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.)

(Esta folha serve somente para guardar o lugar da verdadeira folha de aprovação, que é obtida após a defesa do trabalho. Este item é obrigatório, exceto no caso de TCCs.)

AGRADECIMENTOS

Aos meus pais Maria Lucia Godois Teixeira e Luiz Carlos Cunha Teixeira que sempre estiveram, de forma incansável, ao meu lado ao longo de toda a minha trajetória. A meu filho Bernardo Livi Teixeira que ainda não imagina o quanto serve de combustível para que eu possa continuar aceitando desafios. Agradeço ao meu orientador Cristiano André da Costa por aceitar conduzir e por todo incentivo ao meu trabalho de pesquisa. As Doutoradas Adriana Vial Roehe e Ana Paula da Silva Fernandes por todo conhecimento e material utilizado neste trabalho. Também quero agradecer à Universidade do Vale do Rio dos Sinos, todos os professores pela elevada qualidade do ensino oferecido e, em especial, ao *staff* da secretaria do departamento de computação aplicada por estar sempre a disposição quando precisei. E por fim, ao importantíssimo apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) ao longo desses anos de pesquisa.

RESUMO

A área de computação aplicada vem contribuindo de forma crescente e assertiva em muitas áreas da saúde. Especificamente, na oncologia, as redes neurais artificiais, um segmento de inteligência artificial, na última década, agregaram objetivamente a previsão e o prognóstico de neoplasias malignas / câncer. No entanto, nessa área, ainda há demandas a serem resolvidas e as tecnologias disruptivas são um ótimo aliado para alcançar os resultados esperados. No campo da neoplasia maligna da mama, o exame de imuno-histoquímica é o mais praticado por seu perfil assertivo no que diz respeito ao estadiamento do paciente. O estadiamento significa avaliar e classificar o grau de dissiminação tumoral, visando um tratamento individualizado a cada paciente. Buscando eliminar os níveis de subjetividade envolvidos no diagnóstico do exame de imuno-histoquímica e com a finalidade de reproduzir a rotina diária dos patologistas, o trabalho propõe o uso de duas redes neurais artificiais: uma *support vector machine* e uma *mask R-CNN*. Dois algoritmos de textura: *local binary pattern* e *haralick* foram utilizados para extração do vetor de recurso utilizado como entrada na rede neural artificial SVM e seus resultados obtidos comparados. O trabalho mostra que comparando somente os algoritmos de textura, o *haralick* apresentou uma melhor acurácia que foi de 81% contra 80% de LBP. O uso do modelo *mask R-CNN* através da técnica de TL apresentou desempenho muito abaixo do esperado para o conjunto de dados de IHQ com mAP para treino de 0.050 e para teste de 0.049.

Palavras-chave: Mamas - Câncer. Inteligência artificial. Inovações disruptivas. Redes neurais (computação).

RESUMO

The area of applied computing has been making an increasing and assertive contribution in many areas of health. Specifically, in oncology, artificial neural networks, a segment of artificial intelligence, in the last decade, objectively aggregated the prediction and prognosis of malignant neoplasms / cancer. However, in this area, there are still demands to be resolved and disruptive technologies are a great ally to achieve the expected results. In the field of malignant breast cancer, immunohistochemistry is the most practiced due to its accurate profile with regard to patient staging. Staging means evaluating and classifying the degree of tumor dissemination, aiming at an individualized treatment for each patient. Seeking to eliminate the levels of subjectivity involved in the diagnosis of immunohistochemistry and in order to reproduce the daily routine of pathologists, the work proposes the use of two artificial neural networks: a support vector machine and an Mask R-CNN. Two texture algorithms: local binary pattern and haralick were used to extract the resource vector used as input into the artificial neural network SVM and their results compared. The work shows that comparing only the texture algorithms, haralick showed a better accuracy, which was 81% against 80% of LBP. The use of the mask R-CNN model through the TL technique performed much less than expected for the IHQ data set with mAP for training of 0.050 and for testing of 0.049.

Palavras-chave: Breast - Cancer. Artificial Inteligence. Disruptive innovation. Neural networks (computation).

LISTA DE FIGURAS

1	Mecanismo de metástase a partir da invasão local	24
2	Transição Epitélio-mesenquimal	27
3	Processo de amostras de tecido	29
4	Arquitetura gráfica de um multilayer perceptron com duas camadas escondidas	31
5	Distâncias entre hiperplanos	35
6	A arquitetura de uma rede neural convolucional	38
7	Exemplo de maxpooling com um filtro 2 x 2 e um stride de 2	40
8	Camada convolucional	41
9	Tensor e feature maps	42
10	Processo de aprendizado tradicional vs transfer learning	45
11	Imagem com as reações imuno-histoquímica	56
12	Imagem com as reações IHQ antes e após pré-processamento	58
13	Key points de 50 x 50 pixels detectados em imagens de 256 x 256 pixel	59
14	Patches de 50 x 50 pixels classificados como alta imunoexpressão	59
15	LBP na vizinhança do pixel	61
16	Histograma de uma imagem com e sem imunoexpressão	61
17	Como a textura é computada usando haralick	62
18	Modelo SVM	62
19	Mask R-CNN <i>framework</i> para o conjunto de dados de IHQ	63
20	Visualização IoU	64
21	Arquitetura Mask R-CNN com hiperparâmetros	65
22	Curva de aprendizado de LBP <i>versus</i> Haralick	67
23	Matriz de Confusão Normalizada de LBP <i>versus</i> Haralick	68
24	Matriz de Confusão de LBP <i>versus</i> Haralick	68
25	Roc Curve de LBP <i>versus</i> Haralick	70
26	Validação da Imagem Inteira de LBP <i>versus</i> Haralick	71
27	Imagem atual vs Imagem com predição <i>bounding box</i>	72
28	<i>Precision-Recall Curve Treino versus</i> Teste	73
29	Validação da Imagem Inteira pelo modelo Mask R-CNN	73

LISTA DE TABELAS

1	Estimativas do CM para triênio 2020/2021/2022	15
2	Fatores de risco da neoplasia	22
3	Principais fatores de prognóstico	25
4	Prognóstico, segundo o perfil imuno-histoquímico	26
5	Comparativo dos trabalhos relacionados	54
6	Resultados de Treino e Teste (<i>cross-validation</i>)	69

LISTA DE ABREVIATURAS

BR Gene	Breast Cancer Gene
CAD	Computer-Aided Diagnosis
CDI	Carcinoma Ductal in situ
CIS	Carcinoma In Situ
CM	Câncer da Mama
ConvNet	Convolutional Network
DA	Data Augmentation
DL	Deep Learning
DNA	Ácidodesoxirribonucleico
ECAD	E-caderina
EGFR	Epidermal Growth Factor Receptor (receptor do fator de crescimento epidérmico)
EMT	Transição Epitélio-Mesenquimal
ERK	Quinase Regulada por Sinais Extracelulares
FC-NN	Fully-Convolutional Neural Network
G	Grau tumoral
GPU	Graphics Processing Unit
GCNN	General Classifier Neural Network
GLCM	Matriz de Co-ocorrência de Nível de Cinza
HE	Hematoxilina e Eosina
HER-2	Receptor 2 do fator de crescimento epidérmico humano
HOG	Histogram of Oriented Gradients
IA	Inteligência Artificial
ICPR	International Conference on Pattern Recognition
IHQ	Imuno-Histoquímica
INCA	Instituto Nacional do Câncer
IoU	Interseção of Union
Ki-67	Proteína Ki-67
LBP	Local Binary Pattern
LIBSVM	Library for Support Vector Machine
mAP	mean Average Precision
mAR	mean Average Recall
Mask RCNN	Mask Region Convolution Neural Networks

MET	Transição Mesenquimal-Epitélio
MLP	Multi-Layer Perceptron
NCAD	N-caderina
OMS	Organização Mundial da Saúde
PPA	Porcentagem de área positiva
RE	Receptor de Estrógeno
ResNet	Residual Neural Network
RNA	Rede Neural Artificial
RNC	Rede Neural Convolutacional
Roc	Receiver operating characteristic
RoI	Region of Interest
RP	Receptor de Progesterona
RPN	Region Proposal Network
SLD	Sobrevida Livre da Doença
SVM	Support Vector Machine
TEM	Transporte Epitélio-Mesenquimal
TCGA	The Cancer Genome Atlas
TCLE	Termo de Consentimento Livre e Esclarecido
TL	Transfer Learning
TMAD	The Stanford Tissue Microarray Database
TNM	Tumor primário, estado Nodal e Metástases à distância
TRH	Terapia de Reposição Hormonal
YOLO	You Only Look Once

LISTA DE SIGLAS

CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
UNISINOS	Universidade do Vale do Rio dos Sinos
UFCSPA	Universidade Federal de Ciências da Saúde de Porto Alegre

SUMÁRIO

1	Introdução	15
1.1	Motivação	16
1.2	Questão de Pesquisa	17
1.3	Objetivos	19
1.4	Etapas do Processo de Pesquisa	19
1.5	Organização Textual	20
2	Fundamentação Teórica	21
2.1	Contextualização do Câncer de Mama	21
2.1.1	Estadiamento	23
2.1.2	Metástase Sistêmica	24
2.1.3	Fatores Prognósticos e Preditivos	24
2.1.4	Transição Epitélio-Mesenquimal	27
2.1.5	Histopatologia	28
2.2	Redes Neurais Artificiais	30
2.2.1	Pré-processamento	31
2.2.2	Máquina de Vetores Suportes	34
2.2.3	Redes Neurais Convolucionais	37
2.2.4	Convolução	39
2.2.5	<i>Feature Maps</i>	41
2.2.6	<i>Pooling</i>	42
2.2.7	<i>Dropout</i>	43
2.2.8	Função Ativação	43
2.2.9	Camada <i>Fully-connected</i>	44
2.2.10	<i>Transfer Learning</i>	45
3	Trabalhos Relacionados	47
3.1	<i>A predictive model for high/low risk group according to oncotype DX recurrence score using machine learning</i>	48
3.2	<i>Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system</i>	48
3.3	<i>A study of the suitability of autoencoders for preprocessing data in breast cancer experimentation</i>	50
3.4	<i>Deep Convolutional Neural Networks Enable Discrimination of Heterogeneous Digital Pathology Images</i>	51
3.5	<i>Detection of breast cancer on digital histopathology images: Present status and future possibilities</i>	52
3.6	<i>Deep Multi-instance Networks with Sparse Label Assignment for Whole Mammogram Classification</i>	53
4	Modelo Proposto	56
4.1	Material e Métodos	56
5	Resultados e Discussões	67

6 Conclusão	75
6.1 Contribuições	76
Referências	77
7 Anexos	82

1 INTRODUÇÃO

Para o Brasil, estimam-se 62.280 casos novos de câncer de mama, para cada ano do triênio 2020-2022, com um risco estimado de 61,61 casos a cada 100 mil mulheres.

Sem considerar os tumores de pele não melanoma, a neoplasia da mama ocupa a primeira posição mais frequente em todas as regiões brasileiras com um risco estimado de 81,06 por 100 mil na Região Sudeste; de 71,06 por 100 mil na Região Sul; de 45,24 por 100 mil na Região Centro-Oeste; de 44,29 por 100 mil na Região Nordeste; e de 21,34 por 100 mil na Região Norte como mostra a Tabela 1.

Tabela 1 – Estimativas do CM para triênio 2020/2021/2022

	Estimativa de Novos Casos					
	Estado			Capital		
	Casos	Taxa Bruta	Taxa Ajustada	Casos	Taxa Bruta	Taxa Ajustada
Norte	1.970	21,34	26,24	940	32,91	38,81
Nordeste	13.190	44,29	43,74	4.730	70,19	60,21
Centro-Oeste	3.760	45,24	40,40	860	54,81	43,90
Sudeste	36.470	81,06	57,41	11.670	99,25	69,39
Sul	10.890	71,16	47,96	1.570	75,87	45,90
Total Neoplasia Maligna da mama	66.280	263,09	-	19.770	333,03	-

Fonte: INCA

Em termos globais, excluindo-se os cânceres de pele não melanoma, o câncer de mama constitui-se no mais frequente e comum tumor maligno entre as mulheres, com uma estimativa, para o ano de 2012, de 1,67 milhão de casos novos diagnosticados, o que corresponde a 25,2% de todos os tumores malignos femininos e a uma taxa de incidência de 43,3/100 mil. É a primeira causa de morte por câncer entre as mulheres, sendo estimadas 522 mil mortes para 2012, o que representa 14,7% de todos os óbitos. Embora tenha uma taxa de mortalidade maior do que qualquer outro câncer (12,9/100 mil), o câncer de mama tem letalidade relativamente baixa, dado que a taxa de mortalidade é menor que um terço da taxa de incidência. É também o mais prevalente, com aproximadamente 8,7 milhões de sobreviventes previstos em 2012 (STEWART, 2014).

Observa-se uma variabilidade na incidência segundo as Regiões do mundo, com taxas variando de 27,0/100 mil na África Central e Ásia Oriental a 92,0/100 mil na América do Norte. Em virtude da sobrevivência mais favorável nas Regiões desenvolvidas, a variabilidade das taxas de mortalidade é menor, 6,0/100 mil na Ásia Oriental a 20,0/100 mil na África Ocidental (FERLAY et al., 2015).

Múltiplos fatores estão envolvidos na etiologia do câncer de mama: idade da primeira menstruação menor do que 12 anos; menopausa após os 55 anos; mulheres que nunca engravidaram ou nunca tiveram filhos (nuliparidade); primeira gravidez após os 30 anos; uso

de alguns anticoncepcionais e TRH na menopausa, especialmente se por tempo prolongado; exposição à radiação ionizante; consumo de bebidas alcoólicas; dietas hipercalóricas; sedentarismo; e predisposição genética (pelas mutações em determinados genes transmitidos na herança genética familiar – principalmente por dois genes de alto risco, BRCA1 e BRCA2) (SIEGEL; MILLER; JEMAL, 2017; SYLVA, 2017; STEWART, 2014)

Nos países de baixa e média rendas, o diagnóstico do câncer de mama ocorre em estágios mais avançados da doença, aumentando a morbidade relacionada ao tratamento, comprometendo a qualidade de vida e reduzindo a sobrevida dos pacientes. No intuito de modificar esse cenário, o controle do câncer de mama tem sido uma das prioridades na agenda da Política Nacional de Saúde do Brasil. Assim, o Ministério da Saúde, por meio da publicação “Diretrizes para a Detecção Precoce do Câncer de Mama no Brasil”, recomenda a identificação da doença em estágios iniciais por intermédio das estratégias de detecção precoce, pautadas nas ações de rastreamento e diagnóstico precoce. A mamografia bienal para as mulheres na faixa etária de 50 a 69 anos é a estratégia de rastreio indicada, enquanto o diagnóstico precoce é formado pelo tripé: população alerta para os sinais e sintomas suspeitos; profissionais de saúde capacitados para avaliar os casos suspeitos; e sistemas e serviços de saúde preparados para garantir a confirmação diagnóstica oportuna e com qualidade (Nacional De Câncer; Gomes Da Silva, 2015).

1.1 Motivação

Um formato de matriz 2D é usado para armazenar oito características morfológicas de cada célula e foram usadas para classificá-las em (JAIN et al., 2014). Nesta matriz, as células são representadas usando colunas e as características são representadas usando linhas. As características extraídas são área, perímetro, média, mediana e desvio padrão nas direções x e y. Após, um *General Classifier Neural Network* (GCNN) é usado para classificar células em: cancerosas e não cancerosas. Um procedimento para a segmentação de imagens coradas com hematoxilina e eosina foi proposto por (BEJNORDI et al., 2017). Sete algoritmos de rede neural artificial profunda classificaram as seções de tecido de nódulos linfáticos de mulheres com câncer de mama e compararam com o diagnóstico de patologistas em um ambiente de diagnóstico.

Treinar o classificador usando características pertencentes a quatro categorias: características citológicas que são tamanho e forma dos nucléolos, características de intensidade, características morfológicas que são úteis para obter a forma e a margem (suave ou irregular) das lesões, e características de textura que estão relacionadas à distribuição padrão dos núcleos no interior do tecido, foi realizado por (IRSHAD et al., 2014). A

abordagem de rede neural não é usada somente para detecção de subtipos de neoplasia maligna (ESTEVA et al., 2017; YU et al., 2016) mas também podem complementar outros métodos de avaliação clínica. Deste modo colaboram para melhorar a compreensão dos patologistas sobre a neoplasia, com isso, há um ganho na resposta ao tratamento do paciente (De Sousa E Melo et al., 2013). Resultados precisos de diagnósticos são obtidos através da integração da patologia computacional e os dados clínicos de paciente, como pode ser observado por (DONG et al., 2014).

Motivado pelo crescente emprego da inteligência artificial na área da saúde e em pesquisar melhores formas de estudo na progressão de determinadas neoplasias buscamos neste trabalho, tópicos de relevância que possuem questões em aberto com a finalidade de propor uma abordagem em conformidade com a assistência médica, mas tendo o paciente sempre como foco principal.

Com o destaque de áreas como visão computacional e redes neurais artificiais, *machine learning* ou *deep learning*, as pesquisas em diversas áreas da saúde com foco no paciente ganharam notoriedade e destaque ao ponto de evidenciarem oportunidades em muitas sub-áreas e patologias. Este trabalho é motivado pela vontade de buscar elucidar a relação comportamental das expressões genéticas na neoplasia da mama a fim de trazer alternativas em cuidados de saúde a todos: profissionais de saúde e pacientes.

A avaliação manual de biomarcadores é altamente suscetível a variações subjetivas de um patologista para outro (KHOSRAVI et al., 2018). Uma avaliação errônea de imagens histopatológicas e a tardia tomada de decisão usando lâminas de tecido contendo milhões de células pode custar a vida do paciente (KHOSRAVI et al., 2018). No objetivo de apoiar o melhor estadiamento para o paciente, este trabalho busca preencher a lacuna referente a subjetividade na avaliação histopatológica para o caso da neoplasia maligna da mama.

1.2 Questão de Pesquisa

Este trabalho propõe, em imagem de exame de imuno-histoquímica, a quantificação da expressão de uma proteína presente na transposição epitélio-mesenquimal: a e-caderina. As proteínas da família das caderinas possuem diversas estruturas e funções, que quando alteradas, impedem o desenvolvimento das células mamárias normais e a manutenção fisiológica do tecido, favorecendo a malignidade da mama (FERNANDES; ROHE, 2016). Com base no fenômeno da transposição epitélio-mesenquimal, neste trabalho foi considerado como alvo a quantificação da ECAD que será representada pela alta expressão da membrana celular. A quantificação da expressividade juntamente com os dados clínicos são utilizados pelos patologistas para definir um estadiamento aos pacientes. Entretanto neste trabalho

estaremos abordando somente a detecção e classificação da quantificação através das imagens de reação imuno-histoquímica.

Portanto para atender as lacunas encontradas no parágrafo anterior e a subjetividade de quantificação exposta foi realizado uma busca em bases acadêmicas a partir de uma string contendo palavras chaves que pudessem retornar artigos que remetessem ao estado da arte relacionados ao tópico de pesquisa e que pudessem atingir o objetivo principal.

Como quantificar a alta expressão celular em imagens de imuno-histoquímica através da detecção e classificação de uma rede neural artificial?

Em seu artigo, (KHOSRAVI et al., 2018), propõe vários métodos baseados em rede neural artificial. Demonstrou uma arquitetura básica de rede neural convolucional; uma *Google Inception* com três estratégias de treinamento; e dois algoritmos *ensemble* de estado-da-arte, *Inception* e *ResNet*. Todos utilizam de imagens de imuno-histoquímica para quatro marcadores de câncer de bexiga e dois marcadores de câncer de mama.

As redes neurais artificiais já foram pesquisadas e implementadas em domínios distintos e não muito distantes para o caso da imuno-histoquímica como mostra (BEJNORDI et al., 2017). Qual é a precisão discriminativa dos algoritmos de DL em comparação com os diagnósticos de patologistas na detecção de metástases em linfonodos em estudos de tecido de mulheres com câncer de mama (BEJNORDI et al., 2017)? Ainda em (BEJNORDI et al., 2017), algoritmos de DL mostraram maior distinção do que um grupo de 11 patologistas em um cenário de diagnóstico simulado com restrição de tempo. Com uma acurácia de 0,994 para o melhor algoritmo versus 0,884 para o melhor patologista, o artigo sugere uma utilidade potencial para os algoritmos de DL para o diagnóstico patológico mas ressalta que é preciso uma avaliação em um ambiente clínico.

Trabalhos como (MACÍAS-GARCÍA et al., 2017) que através de modelos de regressão identificam biomarcadores afim de mostrar como a variabilidade intra-observador e interobservador pode afetar nos dados de imuno-histoquímica. Para tal, compara três técnicas de pré-processamento diferentes e aplica aos dados originais, usando uma delas (binarização automática) como baseline de comparação. Ainda em (MACÍAS-GARCÍA et al., 2017), os resultados mostraram que os marcadores imuno-histoquímicos obtidos melhoraram os dados para análise de sobrevivência no câncer de mama. E que o pré-processamento por biomarcadores pode melhorar os bancos de dados de imuno-histoquímica.

1.3 Objetivos

Com o objetivo de agregar e integrar conceitos do estado da arte em redes neurais artificiais para prognóstico clínico a partir de imagens de lâminas que sofreram reação imuno-histoquímica, o trabalho explora a literatura e os estudos recentes relacionados ao tema em cooperação das mais variadas formas. Procura identificar desafios e abrir questões de interesse na área trazendo benefícios no que diz respeito a gestão clínica de pacientes com diagnóstico positivo para neoplasia maligna da mama.

O objetivo principal do trabalho é a partir das imagens imuno-histoquímica propor uma abordagem de rede neural artificial que possa quantificar a positividade da proteína e-caderina, representada pelas regiões onde a reação está mais expressada, através de processos de detecção e classificação. Como garantia, foram criados alguns objetivos específicos:

- Realizar pesquisas para entendimento do que está sendo feito envolvendo redes neurais artificiais e imagens de exames de imuno-histoquímica;
- Procurar e identificar desafios relacionados ao tema;
- Comparar estudos relacionados ao tema;
- Propor uma abordagem de rede neural artificial que atenda a questão de pesquisa;
- Implementar o protótipo da abordagem proposta;
- Executar testes e avaliar os resultados.

1.4 Etapas do Processo de Pesquisa

O trabalho foi dividido em 5 etapas e iniciado no instante em que foi identificado as questões de pesquisa:

- **Conhecimentos:** Consiste em revisão e investigação das literaturas médica e de inteligência artificial ligadas ao tema. Esta etapa forneceu um conhecimentos aprofundado sobre a área do problema.
- **Trabalhos Relacionados:** Etapa em que buscas em bases acadêmicas são feitas com o objetivo de identificar trabalhos similares ao tema proposto.
- **Modelagem:** Etapa em que é proposta uma abordagem de rede neural artificial para solução da lacuna encontrada.
- **Implementação:** Desenvolvimento de um pipeline referente a abordagem proposta.
- **Simulação e Resultados:** Etapa onde demonstra a abordagem proposta e os resultados obtidos.

1.5 Organização Textual

Esta proposta está organizada em quatro seções principais. Inicialmente, um capítulo de Conhecimentos apresenta conceitos básicos para compreensão do restante do trabalho. Seguido de um capítulo Trabalhos Relacionados que apresentam os estudos relacionados ao tema de pesquisa da proposta e esclarece o estado da arte existente, além de abordar e identificar problemas e desafios na área e, também, apresenta respostas as nossas questões de pesquisa. Em seguida, um capítulo Modelo Proposto, alcançando os objetivos do trabalho, apresenta a abordagem proposta que preenche lacunas identificadas no capítulo anterior. Por fim, o capítulo de Metodologia apresenta o fluxo de trabalho empregado.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, abordaremos uma visão geral sobre as duas áreas envolvidas neste trabalho, patologia clínica e redes neurais artificiais. O objetivo é que este capítulo possa fundamentar a abordagem proposta nos capítulos seguintes.

2.1 Contextualização do Câncer de Mama

A neoplasia maligna da mama é uma doença causada pela multiplicação desordenada de células da mama. Esse processo gera células anormais que se multiplicam, formando um tumor. Há vários tipos de câncer de mama, por isso, a doença pode evoluir de diferentes formas. Alguns tipos tem desenvolvimento rápido, enquanto outros evoluem mais lentamente. Depois da neoplasia de pele não melanoma, o câncer de mama é o tipo de doença mais comum entre as mulheres no Brasil e no mundo, e não possui somente uma causa. A idade é um dos mais importantes fatores de risco para a doença, cerca de quatro em cada cinco casos ocorrem após os 50 anos, mas outros fatores podem aumentar o risco da doença como mostra a tabela 2 apresentada pela (SOCIETY, 2013).

Muitos fatores de risco não modificáveis de carcinoma mamário devem ser investigados, como: idade, história familiar, idade da primeira gestação, primeiro fluxo menstrual precoce, menopausa tardia e densidade da mama. Porém, os fatores de risco não modificáveis que merecem destaque, isto é, aqueles de maior importância, são os reprodutivos associados com: fluxo menstrual precoce, idade tardia da primeira gestação, menopausa em idade avançada e presença de doença proliferativa da mama (WILLETT et al., 2000). Algumas mutações resultam em um maior risco como apresenta a Tabela 2. A predisposição genética/hereditária é responsável por apenas 5% a 10% dos casos de câncer de mama, particularmente os genes breast cancer 1 (BRCA1) e breast cancer 2 (BRCA2) estão relacionados com o aparecimento da doença (Martins Da Rosa; RADÜNZ, 2012). Mutações da linhagem germinativa de BRCA1 têm sido detectadas na maioria dos casos de câncer hereditário de mama e ovário. Aproximadamente 1/3 dos casos de carcinoma mamário hereditário e aproximadamente, 30-40% dos esporádicos estão associados com expressão alterada de BRCA1 (MASUDA et al., 2016).

Apesar de ser considerado infrequente em mulheres com menos de 40 anos, a taxa de mortalidade do carcinoma mamário, nesta faixa etária, apresenta-se bem mais elevada, quando comparada as de pacientes peri-menopausicas (CANCELLO et al., 2010). Recentemente foi demonstrado, através de análises genômicas, que os tumores nas pacientes jovens representam uma entidade biológica distinta, caracterizada por padrões moleculares únicos e um pior prognóstico (ANDERS et al., 2008). Frequentemente, nestas pacientes, esta neoplasia

Tabela 2 – Fatores de risco da neoplasia

Risco	Fator de Aumento
> 4.0	<ul style="list-style-type: none"> ● Idade (65+ vs. <65 anos, embora o risco aumente em todas as idades até os 80 anos) ● Hiperplasia atípica confirmada por biópsia ● Certas mutações genéticas herdadas para mama câncer (BRCA1 e / ou BRCA2) ● Carcinoma lobular in situ ● Mamografia de seios densos ● História pessoal de início precoce (<40 anos) de câncer de mama ● Dois ou mais parentes de primeiro grau com câncer de mama diagnosticados em idade precoce
2.1 - 4.0	<ul style="list-style-type: none"> ● História pessoal de câncer de mama (mais de 40 anos) ● Níveis altos de estrogênio ou testosterona endógenos (pós-menopausa) ● Alta dose de radiação no tórax ● Um parente de primeiro grau com câncer de mama
1.1 - 2.0	<ul style="list-style-type: none"> ● Consumo de álcool ● Herança judaica Ashkenazi (Europa Oriental) ● Exposição ao dietilestilbestrol (DES) ● Menarca precoce (<12 anos) ● Alto nível socioeconômico ● Idade avançada na primeira gravidez a termo (> 30 anos) ● Menopausa tardia (> 55 anos) ● Nunca amamentou uma criança ● Obesidade (pós-menopausa) / ganho de peso em adultos ● História pessoal de endométrio, ovário, ou câncer de cólon ● Uso recente e a longo prazo do hormônio da menopausa terapia contendo estrogênio e progesterona

Fonte: INCA

apresenta um maior grau histológico, maior invasão linfovascular, receptores de estrógeno (RE) e progesterona (RP) negativos e aumento da expressão de receptor tipo 2 do fator de crescimento epidérmico humano (HER2), maiores chances de recidiva local e sistêmica, e menor taxa de sobrevida global (CANCELLO et al., 2010).

Estudos sobre o comportamento biológico destes tumores em mulheres jovens e de origem africana apresentam resultados controversos, assim como, estudos abordando novos marcadores patológicos e/ou prognósticos (JONES et al., 2014). Entre esses últimos, encontram-se a expressão imuno-histoquímica da citoqueratinas basais, vimentina e BRCA1 e a pesquisa de alterações genômicas (PARIKH et al., 2008).

Outros fatores de risco são os modificáveis que estão associados a obesidade, pós-menopausa, uso combinado de estrogênio e progesterona na menopausa, consumo de álcool e sedentarismo. Estudos relataram que pacientes que nunca tinham nenhum tipo de

câncer de apresentaram hábitos comportamentais mais saudáveis do que as sobreviventes de carcinoma mamário. Tais como: realização regularmente do exame de mamografia, teste papanicolau, colonoscopia e eram menos propensas a fumar. A prática de atividade física reduz o risco de recidiva do câncer e as taxas de mortalidade; o tabagismo, obesidade e consumo de álcool, não só elevam o risco de carcinoma mamário e recidiva, como favorecem o surgimento de um segundo carcinoma mamário primário (HOMAN et al., 2016).

O tratamento do câncer de mama depende da fase em que a doença se encontra (estadiamento) e do tipo do tumor. Pode incluir cirurgia, radioterapia, quimioterapia, hormonioterapia e terapia biológica (terapia alvo).

2.1.1 Estadiamento

O estadiamento de um caso de câncer significa avaliar e classificar seu grau de disseminação tumoral, visando um tratamento mais adequado para cada paciente, uma vez que pacientes com o mesmo tipo de câncer podem ter diferentes estadiamentos e consequentemente, diferentes tratamentos.

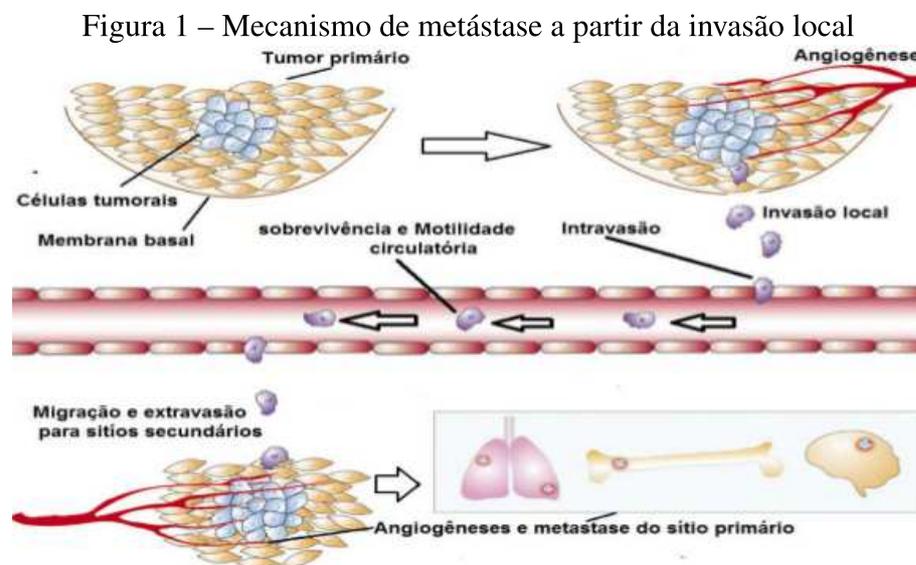
Por mais de 50 anos, a classificação de estadiamento (TNM) tem sido a forma padrão de mensurar a extensão da doença em pacientes com câncer (CHEN et al., 2016). A classificação TNM, baseia-se na extensão anatômica da doença refletida nas características do tumor primário (T), nas características dos linfonodos das cadeias de drenagem linfática do órgão em que o tumor se localiza (N) e na presença ou ausência de metástase a distância (M). Estes parâmetros são graduados em T0 a T4, N0 a N3, e M0 a M1 (INCA, 2011).

O tamanho do tumor está diretamente associado a presença e número de linfonodos axilares envolvidos. Este último é o mais importante indicador de prognóstico para os pacientes com carcinoma mamário em estágio inicial através do envolvimento dos gânglios axilares linfáticos. Além disso, há uma relação direta entre o número de linfonodos axilares comprometidos e o risco de metástase sistêmica.

Muitos avanços vêm ocorrendo no tratamento do câncer de mama nas últimas décadas. Há hoje mais conhecimento sobre as variadas formas de apresentação da doença e diversas terapêuticas estão disponíveis. Quando a doença é diagnosticada no início, o tratamento tem maior potencial curativo. No caso de a doença já possuir metástases (quando o câncer se espalhou para outros órgãos), o tratamento busca prolongar a sobrevida e melhorar a qualidade de vida.

2.1.2 Metástase Sistêmica

A principal causa de morbidade dos pacientes com câncer está atribuída a metastização que é a capacidade de as células cancerosas escaparem da massa do tumor primário e colonizarem locais distantes. As células tumorais sofrem mutação e adquirem motilidade a invasão e começam a invadir tecidos vizinhos, é o início da metastização. Após, essas células metastizantes podem entrar na circulação, diretamente ou através do sistema linfático de locais distantes, permanecendo latentes ou gerando metástase clinicamente detectáveis. A Figura 1 apresenta o mecanismo da metástase do CM para os ossos, pulmão, pleura e fígado (MA; WEINBERG, 2008).



Fonte: Fernandes, Ana Paula da Silva and Rohe, Adriana (2016)

Quando há evidências de metástases (doença a distância), os fatores prognósticos e preditivos são importantes para definir o tratamento que tem por objetivos principais prolongar a sobrevida e melhorar a qualidade de vida do paciente.

As modalidades de tratamento do carcinoma mamário podem ser divididas em:

- Tratamento Local: cirurgia e radioterapia (além de reconstrução mamária);
- Tratamento Sistêmico: quimioterapia, hormonioterapia e terapia biológica.

2.1.3 Fatores Prognósticos e Preditivos

O fator preditivo está associado a resposta do paciente a uma determinada terapia (Cianfrocca e cols., 2004; Bradley 2007) e são universalmente utilizados no carcinoma

mamário. O prognóstico do câncer de mama depende da extensão da doença (estadiamento), assim como das características do tumor.

Segundo a Sociedade Brasileira de Oncologia – SBOC, a tabela 3 apresenta os principais fatores prognósticos independentes que são: estado geral do paciente mensurado pela performance status (medidas pelas escalas ECOG ou Karnofsky), o sítio de metástase, o número de sítios de acontecimento metastáticos, a presença de receptores hormonais e da hiperexpressão de HER2, o intervalo entre o final da adjuvância e o diagnóstico da doença metastática, o uso prévio de quimioterapia adjuvante ou paliativa e o subtipo histológico/molecular do carcinoma mamário.

Tabela 3 – Principais fatores de prognóstico

Fator Prognóstico	Favorável	Desfavorável
Performance status	Bom	Ruim
Sítios de doença	Ossos, partes moles	Vísceras
Números de sítios de doença	Poucos	Múltiplos
Receptores hormonais	Positivos	Negativos
HER2	Negativo	Positivo
Intervalo livre da doença	> 2 anos	< 2 anos
Quimioterapia adjuvante	Não	Sim
Quimioterapia paliativa	Não	Sim

Fonte: Sociedade Brasileira de Oncologia – SBOC.

O grau histológico (G) também é um importante fator determinante de prognóstico, o que foi comprovado em estudos que evidenciaram uma leve correlação entre o grau histológico do CM e a sobrevida dos pacientes. O sistema de classificação de Nottingham (SCN), é o mais utilizado atualmente, foi validado por diferentes pesquisadores e combinado do grau histológico avalia o percentual de diferenciação tubular, avaliação do pleomorfismo nuclear e o índice mitótico. A soma dos três componentes determina os graus: G1 – bem diferenciado, G2 – moderadamente diferenciado, e G3 – fracamente diferenciado (HAMMOND et al., 1999).

A análise dos genes envolvidos com o carcinoma mamário juntamente com a tecnologia de microarray de DNA foi possível classificar os tumores de mama de acordo com seu perfil gênico intrínseco, nas seguintes denominações: Luminal A RE+ e/ou RP+, Luminal B HER2+, Luminal B HER2-, HER2+ não luminal, Basal ou triplo negativo (maior risco de recidiva e menor taxa de sobrevida em 5 anos). A tabela 4 mostra o prognóstico, segundo o perfil imuno-histoquímico a partir do perfil gênico intrínseco.

Todos os perfis são definidos pela expressão de marcadores imuno-histoquímicos e traduzem diversas características em relação ao comportamento biológico da doença e tem

Tabela 4 – Prognóstico, segundo o perfil imuno-histoquímico

Melhor Prognóstico
Luminal A: RE+ e/ou RP HER2- Ki-67 < 14%
Luminal B HER2+: RE+ e/ou RP+ HER2+ Ki-67-
Luminal B HER2-: RE+ e/ou RP+ HER2- Ki-67 > 14%
Menor Prognóstico
HER2+ (não luminal): HER2+ HER2+ (expresso ou amplificado) RE- e RP-
Pior Prognóstico
Triplo - (ductal) RE-, RP-, HER2-

Fonte: Fernandes, Ana Paula da Silva and Rohe, Adriana (2016)

valor prognóstico e preditivo. Os marcadores IHQ: RE, PR, HER2 e Ki-67 apresentam uma importância clínica, pois, sua avaliação definem a melhor opção de tratamento.

Estudos já comprovaram que a presença de receptores hormonais é um fator preditivo poderoso na probabilidade de benefício de terapia adjuvante. A presença de RE e RP, nas células cancerígenas, está correlacionada com o baixo grau histológico e a capacidade de resposta positiva a tratamentos quimioterápicos, especialmente em mulheres na pós-menopausa. Estudos também mostraram que a variação de expressão e correlação entre esses dois receptores corresponde na terapêutica aplicável aos pacientes com carcinoma mamário determinando se melhor ou pior a taxa de sobrevivência.

O receptor HER2 codificado pelo oncogene c-erbB-2 tem um valioso valor preditivo em relação aos benefícios clínicos, pois, amplificado indica a agressividade tumoral e uma maior chance de metástases. Ou seja, HER2 hiperexpressiva indica um prognóstico mais agressivo para o paciente.

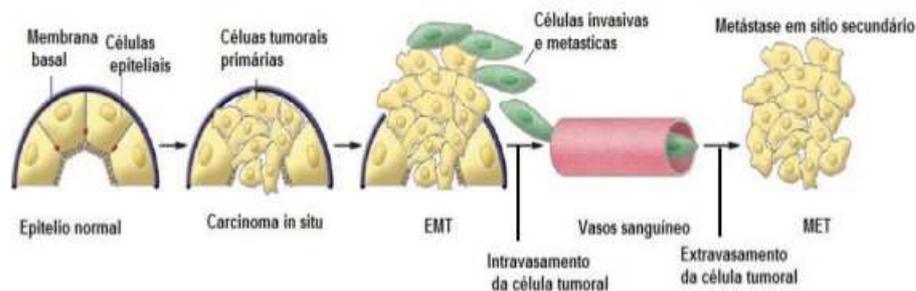
O Ki-67 é um marcador de proliferação celular e está presente em inúmeras neoplasias, estabelece uma correlação com o alto ou baixo grau histológico. Associa-se as altas taxas de proliferação celular, favorece um menor tempo de sobrevivência livre da doença, é um importante biomarcador utilizado na patologia clínica com aplicações potenciais em prognóstico (SUN et al., ; KYLE; RAJKUMAR, 2007).

Esforços tem se concentrado na compreensão de como um tumor primário invade e dissemina-se, utilizando o mecanismo mediado pela mudança fenotípica de epitelial para mesenquimal, transição epitélio-mesenquimal (EMT).

2.1.4 Transição Epitélio-Mesenquimal

Entre os fatores que buscam elucidar aspectos da sua carcinogênese está o estudo da EMT, Figura 2. Esta permite a proliferação de células tumorais, possibilitando maior invasão tecidual e metastização. As proteínas E-caderina e N-caderina atuam na EMT e alterações na sua expressão podem favorecer a progressão tumoral. Hipoteticamente a EMT permite que as células epiteliais tumorais percam sua polaridade celular e sua interação adesiva nas junções célula-célula (GUNASINGHE et al., 2012; ALESKANDARANY et al., 2014). Acredita-se que concomitante ao mecanismo da EMT, ocorram alterações caracterizadas pela mudança do fenótipo epitelial para mesenquimal, acarretando perda ou reduzida expressão de marcadores epiteliais, tais como ECAD e Claudina, e elevada expressão de marcadores mesenquimais, como NCAD, Vimentina, e Twist (DEMIRKAN, 2013; CHOI et al., 2013; STEINESTEL et al., 2014). A EMT pode ocorrer de três formas com diferentes configurações biológicas e distintas funcionalidades: A) tipo I, estão associados com a implantação, formação do embrião e organogênese. Neste processo as células mesenquimais têm potencial fisiológico de sofrer a transição de mesenquimal para epitelal (MET) gerando epitélios secundários; B) tipo II, está associado com a cicatrização de feridas, regeneração de tecidos e a fibrose, auxiliando no mecanismo do reparo tecidual. Normalmente envolve fibroblastos e outras células relacionadas, a fim de reconstrução tecidual; C) tipo III, ocorre em células neoplásicas que sofreram alterações genéticas e epigenéticas, especificamente em genes que favorecem a excrecência clonal e o desenvolvimento de tumores.

Figura 2 – Transição Epitélio-mesenquimal



Estudos analisaram a relação de marcadores epiteliais e mesenquimais com a evolução do tumor, observou que a perda da ECAD está associada aos carcinomas mais agressivos: com fenótipos HER2+ e triplo negativo (CHOI et al., 2013). Acrescenta ainda, que a expressão de marcadores mesenquimais concomitantes à perda da ECAD, é maior em carcinomas invasivos quando comparados aos intraepiteliais.

Estudar o perfil de expressão das proteínas ECAD e NCAD, envolvidas na EMT, através da técnica de IHQ, para o caso de pacientes femininas com diagnóstico de CM é o objetivo desse trabalho, pois, esse estudo necessita de uma melhor elucidação. Essa correlação já foi analisada em outros estudos que não envolveram o uso das redes neurais convolucionais.

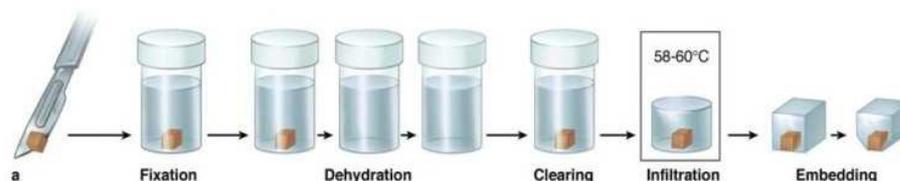
2.1.5 Histopatologia

A histopatologia é o estudo morfológico dos tecidos doentes. No caso do carcinoma da mama refere-se ao exame de uma amostra de biópsia que é artificialmente tirada e separada em lâminas microscópicas para estudar o crescimento do câncer, além da progressão genética, a morfologia celular para encontrar e antecipar tumores. O principal objetivo da maioria das técnicas digitalizadas é obter dados quantitativos de imagens. Esses dados quantitativos incluem o tamanho da célula, anormalidades no tecido e número desproporcional de células. A avaliação apenas visual de padrões histopatológicos bem estabelecidos é tipicamente lenta e mostra-se imprecisa e improdutivo em certos casos de diagnóstico de subtipos e estágios de tumores (MOSQUERA-LOPEZ; ESCOBAR; AGAIAN, 2015).

Como mostra a Figura 3, o processamento de amostras de tecido após a biópsia segue as seguintes etapas: Fixação, onde a amostra através de fixador e formalina passa por um processo de alteração física e química; Desidratação, onde a água na amostra é substituída progressivamente pelo álcool; Limpeza, processo de retirada do álcool que é imiscível a cera; Infiltração por cera, o tecido é infiltrado com uma cera histológica adequada; e Incorporação por bloco, tecido completamente infiltrado é transformado em um "bloco" que pode ser fixado em um micrótomo para o corte da seção.

A avaliação das lâminas de histopatologia microscópica é atualmente o procedimento padrão para estabelecer um diagnóstico e identificar os subtipos de diferentes tipos de câncer e conseqüentemente ter uma melhor resposta no que se refere a tumor, linfonódolos, metástase, recidiva e morte. Um dos principais desafios da patologia computacional é que as imagens de tecidos de tumores geralmente variam em cores e em escala de efeitos de lote em diferentes laboratórios de pesquisa e instalações médicas devido a diferenças nos métodos de preparação de tecidos e implementos de imagem (KHOSRAVI et al., 2018).

Figura 3 – Processo de amostras de tecido



Fonte: Sociedade Brasileira de Oncologia – SBOC.

Após a revisão histopatológica de todo material proveniente das biópsias e/ou peças cirúrgicas para confirmação diagnóstica e seleção das áreas mais representativas do tumor, procede-se ao exame da IHQ para avaliar a expressão de proteínas e/ou carboidratos modificados.

A IHQ tem uma contribuição relevante para o diagnóstico assertivo e o sucesso do tratamento. É um método de análise dos tecidos via microscópio, buscando identificar características moleculares das doenças. Para um melhor estadiamento é importante a avaliação dos seguintes marcadores: ER, PR, Her-2 e Ki-67 (FERNANDES; ROHE, 2016). A perda da proteína ECAD está associada aos carcinomas mais agressivos e acrescenta ainda, que a expressão de marcadores mesenquimais concomitantes à perda da ECAD, é maior em carcinomas invasivos quando comparados aos intraepiteliais (FERNANDES; ROHE, 2016).

Em seu trabalho, (FERNANDES; ROHE, 2016), analisando e quantificando a expressão da proteína ECAD obteve dados quantitativos sobre a expressão dos anticorpos. Para tal, foi utilizado software específico chamado ImageJ, onde imagens de cor castanha, que representam áreas de IHQ positivas, através de método de convolução de cor foram transformadas em imagens binárias. O método destacou imagens RGB, separando a hematoxilina, o DAB e as cores complementares. A imagem foi processada permitindo a quantificação da imunopositividade através de uma ROI de 50 x 50 pixels. A ROI foi posicionada em quatro regiões da imagem binária, com foco nas regiões com maior positividade IHQ. As ROI foram lidas utilizando o software ImageJ, e os resultados foram registrados como percentagem de ROI marcadas por transformação binária. Cada imagem foi dividida em 4 novas imagens e para cada uma, quatro medidas foram tomadas, resultando em 16 medidas. Estas imagens foram analisadas como histograma e importadas para uma planilha (Excel 2007, Microsoft Office, CA, EUA). O histograma especifica dados sobre o número e a intensidade de pixels, permitindo a quantificação da quantidade de DAB em cada imagem, através de porcentagem.

As variáveis quantitativas foram registradas como média e desvio padrão ou mediana e intervalo interquartil.

O cenário atual da IHQ proporciona, pela automação, maior controle na qualidade do processo. A digitalização das imagens e os softwares computacionais já oferecem a possibilidade de análises quantitativas, várias das quais já geraram pesquisas para validação no contexto clínico.

As abordagens de IA podem ser complementares a outros métodos de avaliação clínica para melhorar o conhecimento dos patologistas sobre a doença e melhorar os tratamentos.

2.2 Redes Neurais Artificiais

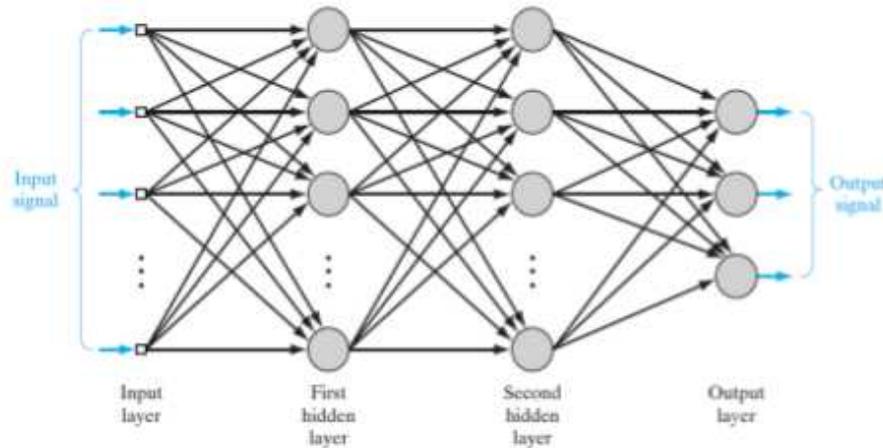
O primeiro modelo de neurônio desenvolvido por McCulloch e Pitts (1943) é uma simplificação do que se sabia na época sobre o neurônio biológico. Para Mitchell (1997, p. 87) sua representação matemática resultou em um modelo com: N valores de entradas; valores em suas arestas, representando sinapses (pesos); apenas 1 valor de saída. Os pesos são multiplicados às entradas e determinam em que nível o neurônio deve considerar os sinais de disparo. O neurônio de McCulloch e Pitts soma o produto das entradas pelos pesos: se esse valor ultrapassar um limiar, ele produz o disparo 1 senão produz o disparo 0.

O *Perceptron* é a forma mais simples de rede neural artificial. É um modelo com um único neurônio, capaz de fazer regressão linear ou separação de duas classes. Embora seja simples e tenha a limitação de fazer somente separações de forma linear é o fundamento básico para construção de grandes modelos RNAs, inclusive modelos convolucionais e recorrentes. Essa limitação, descreve Haykin e Haykin (2009, p. 126), mudou com o algoritmo de *backpropagation* e o surgimento do *multi-layer perceptron* que consegue resolver problemas não-linearmente separáveis a partir da função de ativação, utiliza uma ou mais camadas ocultas como mostra a Figura 4.

Ainda para Mitchell (1997) podem ser: totalmente (do inglês *fully*) ou semi conectada. Na Semi nem todos os nós de uma camada para outra estão interligados, e na totalmente todos os nós de uma camada para a seguinte estão conectados. O gradiente descendente é a base do algoritmo de *backpropagation*, possui algumas variações como: *backpropagation with momentum* e *stochastic backpropagation*.

Para Rezende (2003) as RNAs são compostas de um conjunto de neurônios e sinapses; as informações ficam implícitas nas sinapses; aprendem através de treinamento e ajuste de pesos

Figura 4 – Arquitetura gráfica de um multilayer perceptron com duas camadas escondidas



Fonte: Neural Networks and Learning Machines, Simon Haykin

de forma empírica; uma vez ajustados os pesos, a rede é capaz de produzir respostas com determinado nível de confiança. São utilizadas para problemas de classificação ou predição. Uma RNA deve aprender um modelo do mundo em que estão inseridas; manter um modelo consistente com o mundo real, de forma a atingir os objetivos de interesse de onde está sendo aplicada. Uma RNA trabalha a partir de informação prévia, ou seja, pela representação dos fatos que dizem o que é pelo que se conhece; e/ou pelas observações do mundo, geradas por medidas provenientes de sensores ou cálculos. Essas observações fornecidas a RNA podem ser com rótulos: onde cada padrão de entrada possui uma resposta desejada conhecida; ou sem rótulos: onde os padrões de entrada possuem diferentes percepções do ambiente por si só que devem ser clusterizadas (agrupadas) pela RNA. De qualquer forma, quaisquer observações apresentam o conhecimento do ambiente para a RNA.

O objetivo deste trabalho é detectar e classificar regiões de alta expressão da membrana celular a partir de imagens de exames de imuno-histoquímica. Entretanto o reconhecimento de padrões abrange os seguintes processos principais: (a) aquisição da imagem; (b) pré-processamento, onde filtragens são opcionais para simplificar a informação. A segmentação é o principal processo desta fase; (c) extração das características e (d) aplicação de algoritmo de classificação ou regressão.

2.2.1 Pré-processamento

Redes neurais artificiais profundas tiveram um desempenho notavelmente bom em muitas tarefas de visão computacional. No entanto, essas redes dependem fortemente de *big data* para evitar o excesso de ajustes. O sobreajuste se refere ao fenômeno em que uma rede aprende uma função com variação muito alta, como para modelar perfeitamente os dados de treinamento. Infelizmente, muitos domínios de aplicações não têm acesso a *big data*, como a

análise de imagens médicas (SHORTEN; KHOSHGOFTAAR, 2019).

Para Shorten e Khoshgoftaar (2019), *data augmentation* é uma solução para o problema de dados limitados, ou seja, um pequeno conjunto de dados. O DA engloba um conjunto de técnicas que aprimoram o tamanho e a qualidade dos conjuntos de dados de treinamento, de forma que melhores modelos de DL possam ser construídos.

Os modelos de DL precisam de muitos dados e/ou maior variabilidade de uma mesma classe para treinar devido a um universo de parâmetros muito grande a ser otimizado. Os métodos de DA podem ser utilizados para geração de uma base de treinamento aumentada. Isso é feito aplicando operações de transformações fotométricas e/ou geométricas em cada imagem do conjunto de dados gerando novas imagens por imagem original existente. Shorten e Khoshgoftaar (2019) apresenta essas técnicas de DA a seguir:

1. *Color space*: Os dados da imagem digital geralmente são codificados como um tensor da dimensão (altura largura canais de cores). A realização de aprimoramentos no espaço dos canais de cores é outra estratégia muito prática de implementar. Os aprimoramentos de cores muito simples incluem o isolamento de um único canal de cores, como R, G ou B. Uma imagem pode ser rapidamente convertida em sua representação em um canal de cores, isolando essa matriz e adicionando 2 matrizes zero dos outros canais de cores. Além disso, os valores RGB podem ser facilmente manipulados com operações simples de matriz para aumentar ou diminuir o brilho da imagem. Aumentos de cores mais avançados vêm da derivação de um histograma de cores que descreve a imagem. A alteração dos valores de intensidade nesses histogramas resulta em alterações de iluminação, como as usadas nos aplicativos de edição de fotos.
2. *Cropping*: Cortar imagens pode ser usado como uma etapa prática de processamento para dados de imagem com dimensões métricas de altura e largura cortando uma área central de cada imagem. Além disso, o corte aleatório também pode ser usado para fornecer um efeito muito semelhante às translações. O contraste entre o corte aleatório e as translações é que o corte reduzirá o tamanho da entrada, como (256,256) (224, 224), enquanto as translações preservam as dimensões espaciais da imagem. Dependendo do limite de redução escolhido para o corte, isso pode não ser uma transformação de preservação do rótulo.
3. *Rotation*: Os aumentos de rotação são feitos girando a imagem para a direita ou esquerda em um eixo entre 1 e 359 . A segurança dos aumentos de rotação é fortemente determinada pelo parâmetro do grau de rotação. Rotações leves, como entre 1 e 20 ou - 1 a - 20, podem ser úteis em tarefas de reconhecimento de dígitos, como MNIST, mas à

medida que o grau de rotação aumenta, o rótulo dos dados não é mais preservado após a transformação.

4. *Translation*: Deslocar as imagens para a esquerda, direita, para cima ou para baixo pode ser uma transformação muito útil para evitar distorções posicionais nos dados. Por exemplo, se todas as imagens em um conjunto de dados estiverem centralizadas, o que é comum em conjuntos de dados de reconhecimento de face, isso exigiria que o modelo também fosse testado em imagens perfeitamente centralizadas. Como a imagem original é traduzida em uma direção, o espaço restante pode ser preenchido com um valor constante, como 0 s ou 255 s, ou pode ser preenchido com ruído aleatório ou gaussiano. Esse preenchimento preserva as dimensões espaciais da imagem pós-aumento.
5. *Noise injection*: Consiste em injetar uma matriz de valores aleatórios geralmente extraídos de uma distribuição gaussiana. A injeção de ruído é testada por Moreno-Barea et al. (2018) em nove conjuntos de dados do repositório Dua et al. (2017). Adicionar ruído às imagens pode ajudar as redes neurais convolucionais a aprenderem recursos mais robustos.
6. *Kernel filters*: Os filtros do kernel são uma técnica muito popular no processamento de imagens para aprimorar e desfocar as imagens. Esses filtros funcionam deslizando uma matriz $n \times n$ por uma imagem com um filtro de desfoque Gaussiano, que resultará em uma imagem mais desfocada, ou em um filtro de borda vertical ou horizontal de alto contraste que resultará em uma imagem mais nítida nas bordas. Intuitivamente, as imagens desfocadas para aumento de dados podem levar a uma maior resistência ao desfoque de movimento durante o teste. Além disso, o aprimoramento de imagens para o aumento de dados pode resultar no encapsulamento de mais detalhes sobre objetos de interesse.

Independente da técnica de DA utilizada, transformações geométricas, transformações de espaço de cores, filtros de kernel, imagens misturadas e apagamento aleatório, é importante ressaltar que quase todas essas transformações também vêm com um parâmetro de magnitude de distorção associado.

Ainda em Shorten e Khoshgoftaar (2019), todos os métodos de aumento discutidos acima são aplicados às imagens no espaço de entrada. As redes neurais artificiais são incrivelmente poderosas no mapeamento de entradas de alta dimensão em representações de menor dimensão. Essas redes podem mapear imagens para classes binárias ou para vetores $n \times 1$. O processamento sequencial de redes neurais artificiais pode ser manipulado de modo que as representações intermediárias possam ser separadas da rede como um todo. As representações dimensionais inferiores dos dados da imagem em camadas totalmente conectadas podem ser extraídas e isoladas.

Uma pergunta interessante para o data augmentation é: como determinar o tamanho ideal do conjunto de dados pós-aumentado? Não há consenso sobre qual proporção do tamanho original resultará em um conjunto de dados final que trará melhor desempenho ao modelo. A ampliação morfológica excessiva do conjunto de dados para um modelo de rede neural profunda pode ocasionar ajuste para pior causando *overfitting*.

Uma consideração importante é o viés intrínseco no conjunto de dados inicial limitado. Não existem técnicas de DA que possam corrigir um conjunto de dados com uma diversidade muito baixa em relação aos dados de teste. Todos esses algoritmos de aumento têm melhor desempenho sob a suposição de que os dados de treinamento e de teste são ambos retirados da mesma distribuição. Se isso não for verdade, é muito improvável que esses métodos sejam úteis (ZHONG et al., 2017).

2.2.2 Máquina de Vetores Suportes

Máquinas de Vetores Suportes (do inglês Support Vector Machine) é uma técnica de aprendizado de máquina supervisionado para classificação capaz de analisar dados e reconhecer padrões. Possui grandes fundamentos teóricos e excelentes sucessos empíricos Vapnik (1998). A máquina de vetores suportes tem como restrição, tornar o peso total da classe positiva igual ao da classe negativa.

A principal motivação que levou Vladimir Vapnik em 1979 a desenvolver a teoria da aprendizagem estatística foi melhorar os problemas relacionados com os super e sub ajustes: seja um conjunto de dados usados para treinar uma SVM. Diz-se que a capacidade da SVM está super ajustada (do inglês *overfitting*) quando ela praticamente memoriza rigorosamente a forma de separar os grupos com base nos dados usados para treino. De outra maneira, a capacidade da SVM está sub ajustada (do inglês *underfitting*) quando ela não consegue aprender características de recursos entre as relações das variáveis.

Este tipo de técnica tem sido aplicado a diferentes tarefas de classificação, como: classificação de texto, reconhecimento de objetos e tarefas de previsão. Há muitas vantagens em usar SVM: quando o número de dimensões é maior que o número de amostras, o subconjunto de usos de pontos de treinamento na função de decisão (chamados de vetores suportes) e a capacidade de aplicar diferentes funções de kernel para a função de decisão Chang e Lin (2011).

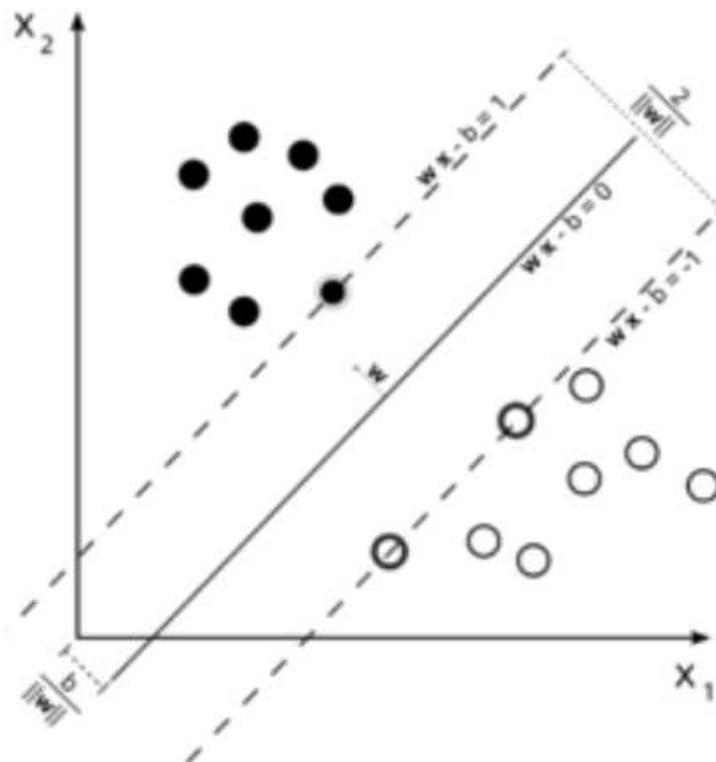
Dado um conjunto de dados usados para treinamento D , composto de n pontos na forma:

$$D = \{(x_i, y_i) | x_i \in \mathbb{R}^p, y_i \in \{-1, 1\}\}_{i=1}^n \quad (2.1)$$

Onde os pontos y_i assumem valor -1 ou $+1$ e indicam a classe a qual os pontos x_i pertençam (problema de separação binário). Cada x_i é de fato um vetor real p -dimensional. O objetivo é encontrar um hiperplano que separe os pontos pertencentes à classe $y = -1$ dos da classe $y = +1$ com a maior margem possível.

Supondo que os dados de treinamento D sejam linearmente separáveis, pode-se escolher dois hiperplanos de maneira que eles separem as duas classes e que não haja nenhum ponto x_i entre eles e então maximizar esta distância entre os dois hiperplanos limítrofes como mostra a Figura 5. A região delimitada pelos hiperplanos também é chamada de margem.

Figura 5 – Distâncias entre hiperplanos



Fonte: Cortes e Vapnik, 1995

Os hiperplanos podem ser descritos pelas Equações 2.2:

$$w \cdot x - b = -1 \text{ e } w \cdot x - b = 1 \quad (2.2)$$

A distância entre os dois hiperplanos é $\frac{2}{\|w\|}$ e como o propósito é a maximização da distância entre eles, o problema torna-se minimizar $\|w\|$. Como é necessário evitar que pontos x_i comecem a avançar na região de margem, as seguintes restrições precisam ser preservadas para todo e qualquer i :

$$w \cdot x - b \geq -1 \quad \text{para } x_i \text{ que pertenam } 1^{\text{a}} \text{ classe } (y_i = 1) \quad (2.3)$$

ou

$$w \cdot x - b \leq 1 \quad \text{para } x_i \text{ que pertenam } 2^{\text{a}} \text{ classe } (y_i = -1) \quad (2.4)$$

As Equações 2.3 e 2.4 podem ser agrupadas da seguinte forma:

$$y_i(w \cdot x - b) \geq 1 \quad \text{para } 1 \leq i \leq n \quad (2.5)$$

Portanto, tem-se um problema de minimização de $\|w\|$, sendo w e b variáveis e restritas à Equação 2.5. Porém a proposta de modificação feita por (VAPNIK, 1998) possibilitou que amostras atípicas fossem acomodadas por um hiperplano fronteiro adaptado a elas. De maneira que classes não linearmente separáveis mas com tendência a separação linear possam ser classificadas por SVM através do ajuste do parâmetro C .

$$w \cdot x - b \geq -1 - \varepsilon_i \quad \text{para } y_i = 1 \quad (2.6)$$

ou

$$w \cdot x - b \leq 1 - \varepsilon_i \quad \text{para } y_i = -1 \quad (2.7)$$

Resolvendo para as restrições impostas pelas Equações 2.6 e 2.7, a nova função de minimização torna-se $\frac{\|w\|^2}{2} + C * \sum (\varepsilon_i)^k$ na qual C é um parâmetro a ser definido pelo usuário e quanto maior o valor de C , maior a penalidade para erros. Em outras palavras, quanto maior o valor de C , mais vetores serão tratados como vetores suportes.

Grande parte dos problemas reais não são possíveis de serem separados pelos classificadores lineares. Portanto, a solução é projetar os dados originais em um novo espaço onde os dados sejam linearmente separáveis. Quando os dados de entrada são conhecidos e o domínio é unidimensional, é relativamente fácil achar uma função de mapeamento. Entretanto, quando se está diante de espaços multidimensionais, esta transformação não é tão imediata.

Este mecanismo de mapear um espaço de entrada em um espaço de características através de uma função $\varphi()$ é chamada de truque de Kernel. Os primeiros kernels a serem testados foram: polinomial, gaussiano e tangente hiperbólico.

Fundamentalmente para casos de classificação, SVM é um classificador binário, ou seja, funciona somente para duas classes mas (CHANG; LIN, 2011) utilizou a biblioteca da LIBSVM para realizar classificação para mais de duas classes. Chang e Lin (2011) mostra casos de sucesso, com uso do LIBSVM, em diversas áreas como: visão computacional, bioinformática, neuroimagem e processamento de linguagem natural. Para casos de regressão e classificação, LIBSVM pode também gerar estimativas de probabilidade.

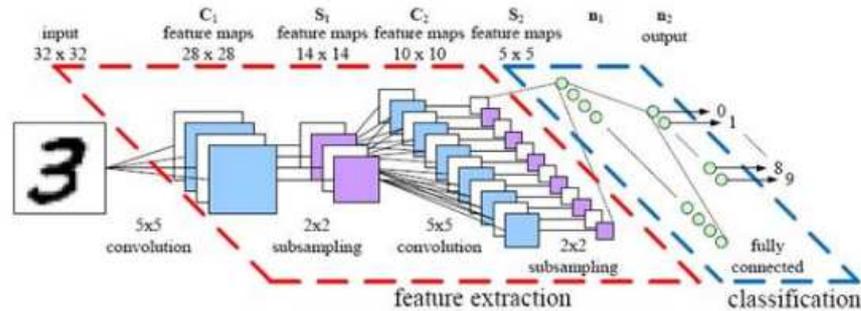
O LIBSVM implementa a abordagem "*one-against-one*" (KNERR; PERSONNAZ; DREYFUS, 1990) para classificação multiclasse. Muitos outros métodos estão disponíveis para a classificação SVM multiclasse. Hsu e Lin (2002) faz uma comparação detalhada e concluiu que a abordagem "*one-against-one*" é a mais competitiva. Além disso, o LIBSVM, diferentemente do SVM que faz apenas predição de rótulos da classe, estende para que o SVM possa fornecer estimativas de probabilidade. Outro fator relevante, é que o LIBSVM provê métodos que fazem a verificação de parâmetros, isto é, os parâmetros com maior precisão de cross-validation são retornados.

2.2.3 Redes Neurais Convolucionais

As RNCs possuem muitos casos de sucesso na área da medicina clínica, alguns deles são: identificação de moléculas com potencial para tratamento de câncer, diagnóstico preciso e rápido de doenças cardiovasculares por MRI, interpretação de imagens médicas usando um conjunto de dados mundial de imagens associadas a termos típicos de diagnósticos, detecção prematura de retinopatia diabética. Al-Masni et al. (2018) demonstra uma detecção e classificação de tumores da mama, quantificação não invasiva da evolução de um tumor, classificação de lesões e tumores. As RNCs são compostas de duas grandes fases: extração de características pelas camadas convolucionais e classificação, como mostra a Figura 6.

As RNCs foram idealizadas no início dos anos 90. De acordo com LeCun, Bengio e Hinton (2015) sua vasta aplicação a partir de 2006 veio devido a "popularização" das GPU's, inspiradas no modelo biológico da visão, isto é, reconhece objetos partindo de primitivas básicas. Ou seja, da menor parte para o todo vai iterativamente construindo conceitos mais abstratos. Tem o conceito de DL, ou seja, apresentam multi-camadas. Seu treinamento requer alto custo computacional necessitando de um conjunto de dados grande o suficiente que atenda essa etapa.

Figura 6 – A arquitetura de uma rede neural convolucional



Fonte: Neural Networks and Learning Machines, Simon Haykin

Uma RNC é basicamente uma RNA tradicional que foi estendida usando pesos compartilhados. Para Zhang, Muhammad e Tang (2018) uma RNC foi projetada para reconhecer imagens, tendo convoluções dentro, que vêem as bordas de um objeto reconhecido na imagem. Uma RNC é uma rede neural artificial MLP desenvolvida especificamente para reconhecer formas de duas dimensões com alto grau de invariância à translação, alteração de tamanho, distorção e outras formas de distorção (HAYKIN; HAYKIN, 2009). Desta forma, a tarefa de reconhecimento de formas ou imagens é aprendida pelo modelo através de um aprendizado supervisionado, dividido nas seguintes fases:

1. Extração de Atributos (do inglês Feature Extraction) - cada neurônio obtém um entrada sináptica de um campo receptivo local da camada anterior, deste modo empreende a extração de atributos locais (ou local features do inglês). A partir do momento que um atributo ou traço é extraído, a localização exata do mesmo torna-se menos importante, de forma que sua posição relativa aos outros atributos ou traços é praticamente preservada.
2. Mapeamento de Atributos (do inglês Features Maps) - Cada camada da rede é composta por múltiplos mapas, de modo que cada mapa de atributos figura tal como um plano, onde neurônios individuais são separados para compartilhar o mesmo conjunto de pesos sinápticos.
3. *Subsampling* - cada camada convolucional é seguida por uma camada que empreende as operações de *local averaging* e *subsampling*, de modo que a resolução do mapa de atributos ou traços é reduzido.

Todos os pesos em todas as camadas de uma RNC são aprendidos através da fase de treinamento, processo em que a rede aprende seus próprios atributos ou traços de forma automática. Ou seja, uma RNC pode vir a exibir múltiplas camadas de convolução e *subsampling*, camadas estas responsáveis por extrair os atributos ou traços mais relevantes para a tarefa de aprendizado. A camada de convolução é composta por diversos neurônios, de modo que cada um é responsável por aplicar um filtro em uma parte específica de uma

imagem, da matriz que representa ou figura como o padrão de entrada.

Desta forma, cada neurônio está conectado a um conjunto de *pixels* da camada anterior e cada uma destas conexões, produz uma saída que é então passada para a camada subsequente. Entretanto é importante notar que os pesos das conexões podem ser interpretados como uma matriz que figura tal como um filtro nas operações de convolução, também conhecida como *kernel*. Os *kernels* são os responsáveis diretos pela extração dos atributos. Em uma RNC a extração do conhecimento ocorre em operações sucessivas de convolução e *subsampling* realizadas em camadas sucessivas, respectivamente.

Por sua vez, os *kernels* geralmente são matrizes quadradas de ordem 3, 5 ou 7. Estas matrizes percorrem toda a matriz de entrada ou imagem, extraindo assim os traços ou assinaturas. Posteriormente, uma camada de *subsampling* é utilizada para reduzir a dimensão do objeto gerado a partir da convolução, imediatamente anterior, como pode ser observado na ilustração contida na figura 5.

2.2.4 Convolução

Goodfellow, Bengio e Courville (2016) afirma que uma convolução é uma operação que envolve duas funções, de modo que em tarefas de aprendizado de máquina, a entrada comumente figura como uma matriz multidimensional e o *kernel* também é uma matriz multidimensional de parâmetros. Logo, tais parâmetros se adaptam no decorrer do aprendizado, tendo seus respectivos valores iniciais escolhidos de forma randômica pelo algoritmo de treinamento.

As RNCs empregam a operação de convolução no lugar da multiplicação matricial padrão (inerente a uma camada tipo *perceptron*) em pelo menos uma de suas camadas (GOODFELLOW; BENGIO; COURVILLE, 2016). Essa operação, que é linear, permite explorar informações em séries temporais ou imagens. A convolução para uma estrutura de série temporal 1-D de duas sequências $x(n)$ e $w(n)$, é representada pela equação $y(n)$ abaixo:

$$y(n) = \sum_{-\infty}^{\infty} x(k)w(n-k) = (x * w)(n) \quad (2.8)$$

Se $x(n)$ for um sinal discreto e $w(n)$ for a resposta ao impulso de um sistema linear e invariante no tempo, pode-se mostrar que a saída desse sistema será exatamente a sequência $y(n)$ como expressa acima. No âmbito de redes, costuma-se dizer que $x(n)$ é a entrada e $w(n)$ é o kernel (GOODFELLOW; BENGIO; COURVILLE, 2016).

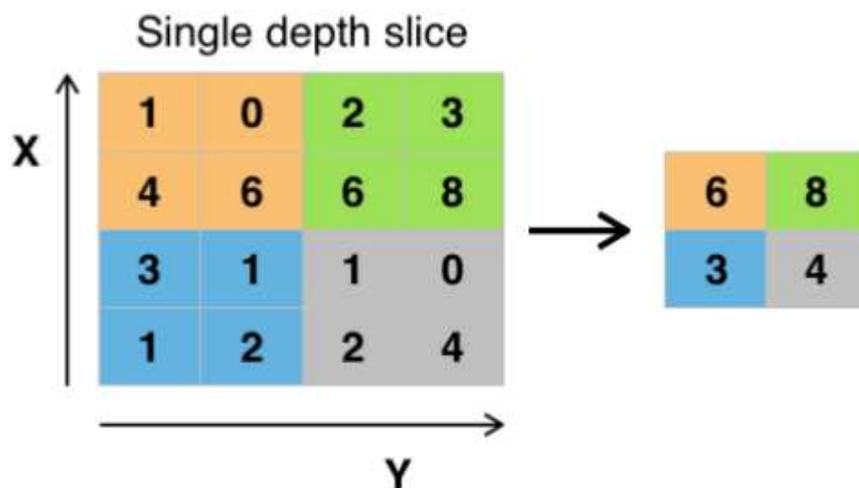
É possível estender a mesma ideia para duas dimensões, ou seja, quando trata-se com imagens. Segundo Goodfellow, Bengio e Courville (2016), uma imagem de duas dimensões como entrada para uma rede convolucional, normalmente será esperado que o kernel também seja uma matriz quadrada, tal que a operação pode ser definida pela equação:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (2.9)$$

O que torna a camada convolucional diferente da camada totalmente conectada (do inglês fully-connected) é que todos os campos receptivos são filtrados com os mesmos pesos locais para todo pixel resultando em saídas ainda no formato bidimensional. Por trabalhar com matrizes multidimensionais com profundidade d é utilizado o termo tensor.

Outro fator importante de mencionar é o passo (do inglês stride), como mostra a figura 7. As convoluções mais “conservadoras” são feitas com passo 1, ou seja, filtra todos os *pixels* então para uma imagem de entrada de tamanho 64×64 é gerada uma nova imagem com o mesmo tamanho. O uso de passos maior que 1 é comum quando o objetivo é reduzir tempo de execução, pulando *pixels* e assim gerando imagens menores. Um passo 2 para imagem de entrada 64×64 gera uma nova imagem de tamanho 32×32 . O passo informa que o *kernel* passará pela matriz de entrada utilizando apenas um passo na horizontal ou um passo na vertical, sem utilizar uma técnica conhecida como zero *padding*, que nada mais é o preenchimento com zeros para completar a matriz de entrada para a operação de convolução.

Figura 7 – Exemplo de maxpooling com um filtro 2×2 e um stride de 2



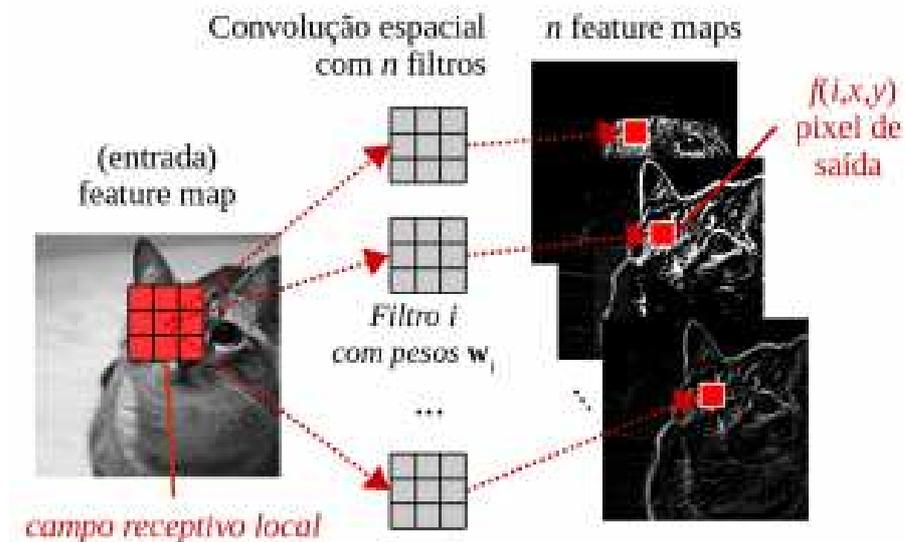
Fonte: Roffo (2017)

Uma imagem inteira (colorida, multicanais e etc) é introduzida como entrada, na camada convolucional cada neurônio é um filtro aplicado a imagem de entrada. O resultado da

convolução é o mapa de ativação ou mapa de características (do inglês *feature maps*). Cada filtro é uma matriz de peso e representa uma região da imagem processada, chamando-se campo receptivo local (do inglês *local receptive field*). Um valor de saída (*pixel*) é uma combinação dos *pixels* de entrada nesse campo receptivo local, como mostra a Figura 8.

Cada filtro é ligeiramente menor que a imagem original por que as bordas são descartadas e pode representar um atributo específico da imagem, p.ex., reta, curva, cor, etc. Sucessivas convoluções detectam atributos de mais alto nível, p.ex., semi-círculos (reta e curva), quadrados (4 retas), etc. Em mais alto nível ainda pode detectar um texto manuscrito ou um objeto de uma cor específica.

Figura 8 – Camada convolucional



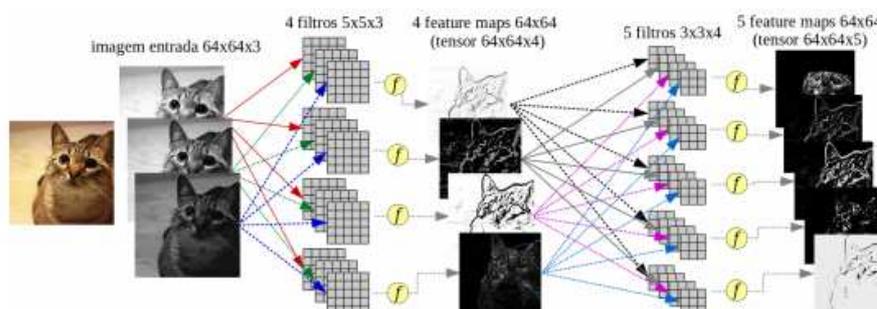
Fonte: Neural Networks and Learning Machines, Simon Haykin

2.2.5 Feature Maps

Os *features maps* gerados pelos filtros da camada convolucional são empilhados, formando um tensor que tem profundidade igual ao número de filtros. Esse tensor serve como entrada para a próxima camada como mostra a Figura 9. Como a primeira camada convolucional gera um tensor $64 \times 64 \times 3$, os filtros da segunda camada terão profundidade 4. No caso de uma terceira camada convolucional, os filtros devem ter profundidade 5.

A fase de extração de características transforma uma imagem em um vetor de características pela sequência de convoluções, ativação e *poolings*. A convoluções alteram o dados e aprendem os filtros; o *pooling*, reduz a escala para a próxima camada. A extração de características é possível através:

Figura 9 – Tensor e feature maps



Fonte: Neural Networks and Learning Machines, Simon Haykin

1. Do cálculo de ajuste automático dos pesos dos filtros, *BackPropagation*, durante o treinamento do modelo;
2. Cada filtro aprendido é um extrator de características;
3. Cada imagem resultante de um filtro é um mapa de características;
4. Os *poolings* são necessários para reduzir a quantidade de características por filtro, isto é, redução de escala.

2.2.6 Pooling

A operação de *pooling* segue após a obtenção dos atributos na etapa de convolução. Esta operação é responsável por trocar os valores da matriz de saída, a mesma obtida após a operação de convolução, por uma representação estatística destes valores. Como ilustração, uma possibilidade tem-se a operação chamada *max pooling*, que efetua o cálculo do maior valor dentro de uma região delimitada por um "retângulo". Este "retângulo" nada mais é que um delimitador de uma área de itens vizinhos da matriz de entrada, que passa pela matriz, efetuando o cálculo e modificando a mesma por esta representação.

O *pooling* também é importante para tornar a representação aproximadamente invariante a pequenas translações da entrada, ou seja, se a entrada for mudada ligeiramente, os valores das saídas (com *pooling*) ficarão majoritariamente iguais. Isso é muito importante para fazer com que a RNC consiga reconhecer padrões em diferentes posições de uma imagem.

Para Graham (2014) a camada de *pooling* ou camada de sub-amostragem (do inglês *downsampling*) é responsável por reduzir a dimensão espacial dos mapas de características, sendo a operação de *max pooling* normalmente empregada. Essa operação de redução de dimensionalidade tem dois objetivos: como o custo computacional é diretamente proporcional a profundidade dos tensores (d), pois, aumenta ao longo das camadas, é recomendável reduzir

a dimensão espacial dos mesmos. Isto é, reduz o número de pesos compactando os dados; a redução do tamanho das imagens para obter uma composição de banco de filtros multiresolução que processa imagens em diferentes espaços-escala.

O mais comum é o uso do *max pooling* – filtro, normalmente 2 x 2, e passo (do inglês *stride*) do mesmo tamanho detectando o maior elemento da região, máximo morfológico, como mostra anteriormente na Figura 8. A motivação está relacionado a alta resposta ao filtro respectivo, pois, não importa tanto a localização exata uma vez que o atributo está na imagem convoluída. Como Sainath et al. (2013), estudos defendem a não utilização do *pooling* em alguns casos mas recomendam aumentar o passo nas convoluções, o que produz o mesmo efeito redutor. Outra vantagem do *pooling* é controlar o sobre-treinamento (do inglês *overfitting*) e consequente perda de generalização.

2.2.7 Dropout

Essa técnica tem como objetivo atenuar o *overfitting* e é usada somente na fase de treinamento. Na fase de propagação dos dados pela rede, desativada de forma aleatória, com probabilidade p , a ativação de neurônios. Este procedimento pode ser considerado redundante, isto é, classifica corretamente mesmo que algumas ativações sejam zeradas. Também, pode ser uma forma de regularização, pois, ao desativar aleatoriamente neurônios, interfere diretamente nos *features maps* gerados a cada época por meio da redução da complexidade das funções.

2.2.8 Função Ativação

Uma RNC como a ConvNet onde sua principal operação é linear não apresentará bons resultados para problemas não-lineares como a extração de características em imagens. Para resolver esse problema, após a camada convolucional em uma ConvNet, é aplicada uma camada de ativação, onde uma função não-linear como a função logística e tangente hiperbólica (*tanh*) são aplicadas para compor a saída.

Para as camadas *fully-connected* ocultas a função de ativação *softmax* não é ideal. O uso de funções sigmoidais como a logística e a tangente hiperbólica é normal em RNAs. Porém, em aprendizado profundo tem sido mais utilizado, como demonstrado por Fred Agarap (2019), para ter um treinamento mais rápido, a função retificadora linear (do inglês *rectified linear function*, ReLU). Superam a sigmoide, *tanh* e outras funções de ativação, pois, possuem treinamento muito mais rápido com desempenho parecido. Também atenuam o efeito de dissipação do gradiente (do inglês *vanishing gradient*) que é a diminuição exponencial do

gradiente ao longo das camadas tornando o aprendizado muito lento.

Existem diversas funções de ativação possíveis, cada qual adequada a um tipo de cenário. Fred Agarap (2019) demonstra a função softmax que é considerada uma função de ativação para classificação, aplicada a camada de saída da rede. Tem por objetivo definir a probabilidade de uma classe dentro de um problema multiclases. Em problema multiclases, normaliza entre 0 e 1 as saídas, que são obtidas por um classificador linear (DENKER; LECUN, 1991) e pode ser definido como:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}; j = 1, \dots, K \quad (2.10)$$

2.2.9 Camada *Fully-connected*

Ao final, como mostra a Figura 5, temos uma camada totalmente conectada, *fully-connected*, que estabelece quais atributos de alto nível, última camada convolucional, mais se correlacionam com a classe do objeto. Por exemplo, a imagem de um cachorro teria resposta alta no mapa de ativação correspondente a patas e rabo, ou um pássaro teria resposta alta para bico e asas.

RNCs possuem a seguinte sequência: camadas convolucionais seguidas por funções de ativação, ocasionalmente utilizam *pooling* ou não, e, posteriormente, uma série de camadas *fully-connected* que também são acompanhadas de funções de ativação.

Na criação de uma RNC é preciso definir os seguintes hiper-parâmetros:

1. O número de camadas convolucionais, isto é, para cada camada é preciso definir o número de filtros, seu tamanho e o tamanho do passo dado durante a convolução;
2. Número de camadas de *pooling*, neste caso é necessário escolher o tamanho da janela e do passo que definirão o fator de subamostragem;
3. O número de camadas *fully-connected* e o número de neurônios contidos em cada uma dessas camadas.

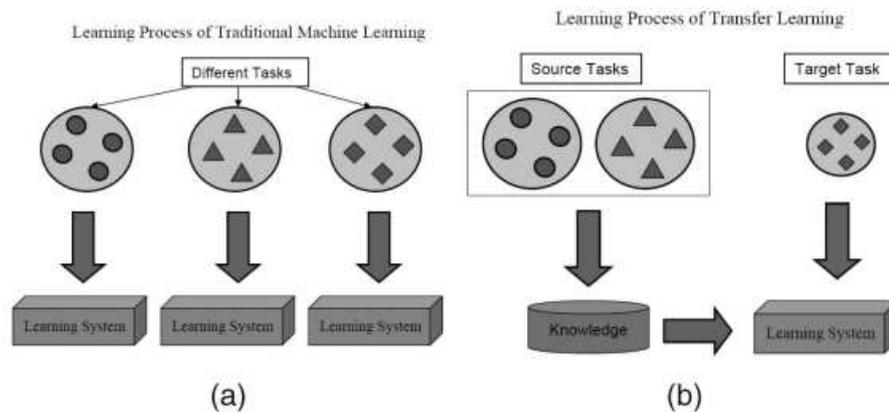
Como mostra Akkus et al. (2017) o número de parâmetros em uma RNC está relacionado aos valores a serem aprendidos em todos os filtros nas camadas convolucionais, os pesos das camadas *fully-connected* e os bias. Isto é, eles podem ser grandes e crescer consideravelmente quanto maior for a arquitetura utilizada.

2.2.10 Transfer Learning

A inicialização dos pesos de uma RNC pode influenciar diretamente na sua capacidade de treinamento. Uma inicialização ruim dos pesos pode fazer com que o algoritmo de otimização inicie em uma região do domínio onde existam vários mínimos locais. Portanto, ao invés de inicializar os pesos com valores aleatórios, é melhor utilizar valores que já tenham sido treinados com outro conjunto de dados. Esse procedimento recebe o nome de *Transfer Learning* como mostra a Figura 10.

TL é uma técnica onde na implementação de uma arquitetura de RNC é possível utilizar uma modelo pré-treinado, pesos e parâmetros, em uma rede, geralmente pública, e finalmente ajustada ou não para uma aplicação específica. A parte treinada funciona como um extrator de características de imagem, os pesos são congelados, isto é, não se alteram durante o *backpropagation* (LONG et al., 2015). Entretanto, vale ressaltar que tal procedimento vai além da inicialização de pesos, permitindo o reaproveitamento da arquitetura de rede original. Um dos benefícios de aplicar TL em modelos de aprendizado profundo é o de trazer mais agilidade/rapidez na implementação dos modelos.

Figura 10 – Processo de aprendizado tradicional vs transfer learning



Fonte: Pan e Yang (2010)

Para Karpathy (2017) existem três situações de destaque na aplicação de TL:

1. Quando o conjunto de dados é grande e semelhante ao utilizado pela RNC já treinada. Nesse caso, pode-se utilizar os pesos da rede original sem a necessidade de efetuar uma etapa de treinamento;
2. Quando o conjunto de dados é pequeno e semelhante ao utilizado pela RNC já treinada. Neste caso, os pesos da rede são inicializados com os valores dos pesos da RNC já

treinada. Usando o novo conjunto de dados, deve-se efetuar uma etapa de treinamento, de maneira a adequar os pesos ao problema trazido pelo novo conjunto de dados. A essa adequação de pesos ao problema trazido, dá-se o nome de *fine tuning*;

3. Quando o conjunto de dados é diferente do utilizado na rede neural previamente treinada. Então nesse caso, deve-se fazer o *fine tuning*.

As camadas iniciais detectam padrões de baixo nível, como curvas e bordas. A última camada da rede pré-treinada é removida e substituída pelo classificador que deseja utilizar. Entretanto, se a base é muito diferente da rede "original" pode-se usar menos camadas pré-treinadas e mais para o problema específico.

As técnicas de TL foram aplicadas principalmente a aplicativos de pequena escala com uma variedade limitada, como problemas de localização baseada em rede de sensores, classificação de texto e classificação de imagem. No futuro, as técnicas de TL serão amplamente usadas para resolver outros aplicativos desafiadores, como classificação de vídeo, análise de redes sociais e inferência lógica (PAN; YANG, 2010).

Ponti e Costa (2018) define aprendizado profundo como RNAs com um grande número de parâmetros e camadas em uma das quatro arquiteturas de rede fundamentais: redes pré-treinadas não supervisionadas, redes neurais convolucionais, redes neurais recorrentes e redes neurais recursivas. O foco deste trabalho, através da textura de imagens de imuno-histoquímica, concentra-se em: realizar operações de filtragem e morfológicas para eliminação de ruído no conjunto de dados fornecido pela equipe médica; detecção das regiões de interesse que são as áreas onde a reação química apresenta-se mais expressada; treinamento e teste das imagens binarizadas através de uma rede neural artificial com objetivo de detectar e classificar as áreas de alta expressão; e apresentação dos resultados encontrados.

3 TRABALHOS RELACIONADOS

A disseminação das RNAs vêm ganhando destaque nos últimos anos devido ao grande avanço de recurso computacional e a área de diagnóstico clínico, uma sub-área da medicina, está em grande evidência por sofrer positivamente com esse grande impacto tecnológico. A avaliação das lâminas de histopatologia microscópica é atualmente o procedimento padrão para estabelecer um diagnóstico e identificar os subtipos de diferentes tipos de câncer. A avaliação apenas visual de padrões histopatológicos bem estabelecidos é tipicamente lenta e mostra-se imprecisa e improdutivo em certos casos de diagnóstico de subtipos e estágios de tumores (MOSQUERA-LOPEZ; ESCOBAR; AGAIAN, 2015).

As abordagens de RNA podem ser complementares a outros métodos de avaliação clínica para melhorar o conhecimento dos patologistas sobre a doença e melhorar os tratamentos. Estudos anteriores mostraram resultados de diagnóstico mais precisos ao integrar informações extraídas da patologia computacional com dados clínicos de pacientes para vários tipos de câncer, como o câncer de próstata, câncer de pulmão, câncer de mama, câncer colorretal e câncer de ovário (KHOSRAVI et al., 2018).

No intuito de preencher tais lacunas no diagnóstico clínico pelo processo de imuno-histoquímica pela análise subjetiva das proteínas e-caderina e n-caderina, foram feitas buscas, utilizando a string abaixo, por artigos relacionados nas seguintes bases acadêmicas: ACM, IEEE e Elsevier.

Search String

```
("medical image"OR "biomedical image") +("immunohistochemical"OR "cadherin"OR "CD44") +("breast cancer"OR "bladder cancer") +("prognosis") +("detect"OR "classify"OR "predict") +("artificial neural networks"OR "convolutional neural networks"OR "deep learning")
```

Com o objetivo de trabalhar somente com artigos que tivessem a mesma abordagem, a seleção dos artigos foi realizada por meio de processos de critérios de exclusão a saber:

1. Critério de exclusão 1: remoção de artigos que não abordam "immunohistochemical" ou acrônimos relacionados;
2. Critério de exclusão 2: remoção de artigos que não abordam "artificial neural networks" ou "convolutional neural networks" ou "deep learning";
3. Critério de exclusão 3: remoção de artigos que não abordam "breast cancer", "cadherin" ou acrônimos relacionados.

As etapas do processo de filtragem foram executadas da seguinte forma: (1) remoção duplicada, (2) critério de exclusão, (3) remoção de impurezas, (4) filtragem por título, (5) filtragem por resumo e, finalmente, (6) filtragem por texto completo. Assim, temos os artigos que serão detalhados a seguir, e ao final uma tabela resumindo cada um.

3.1 *A predictive model for high/low risk group according to onco-type DX recurrence score using machine learning*

Kim et al. (2019) sugere um score de recorrência de neoplasia da mama que envolve 21 genes que auxilia na tomada de decisão para quimioterapia no estágio inicial de receptor hormonal positivo. O paper utilizou prontuário eletrônico do paciente para acessar dados de patologia clínica e dois scores de recorrência: RS < 11 então baixo risco; RS > 25 então alto risco. Usou 208 casos pacientes como grupo de modelagem e 76 como grupo de validação. Usou método decision jungle para predição de RS alto encontrando 90,30% e 91,70% para acurácia e área abaixo da curva roc, respectivamente. E para predição do RS baixo, usou o método neural networks, encontrou 72,60% e 74,40% para acurácia e área abaixo da curva roc. A validação interna pelos 76 pacientes do grupo de validação, apresentou os seguintes percentuais de RS alto e RS baixo, respectivamente: 88% e 79%. A presença de carcinoma intraductal foi a variável que mostrou diferença entre os dois grupos. A EIC representa mais de 25% da área de todo o carcinoma invasivo. É um fator de risco muito importante para a recorrência da doença após terapia conservadora. Os critérios ODX possuem dependência de uma classificação histológica precisa e resultados imunohistoquímicos de alta qualidade. Foi utilizado um nomograma para cálculo da probabilidade de uma pontuação ODX de alto ou baixo risco para cada paciente. O modelo representa uma ferramenta útil para RS alto mas possui um baixo número de pacientes envolvidos e de mesma etnia.

3.2 *Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system*

Al-Masni et al. (2018) apresenta um novo sistema CAD que utiliza uma RNC baseada em YOLO. O sistema CAD proposto incorpora uma abordagem RNC baseada em ROI que utiliza as camadas convolucionais seguidas por redes neurais totalmente conectadas para detectar a localização adequada da massa e para distinguir os tipos de tumores: benignos ou malignos. Portanto, a proposta CAD contém quatro estágio principais: pré-processamento de mamografias, *feature extraction* utilizando multi convolutional *deep layers*, detecção de massa com modelo de confiança e, por último, a classificação usando uma FC-NN. Foi utilizado um banco de dados de mamografia, muito utilizado em proposta de pesquisa de mama, criado pela

universidade da Flórida para treinar e testar a rede. São coletadas quatro mamografias par cada caso, com duas visões diferentes: oblíqua mediolateral (MLO) e craniocaudal (CC) e cada uma contém suspeita de lesão associada a informações de verdade básica. O uso de 600 mamografias justifica a utilização da técnica de data augmentation e após, todas as mamografias (originais e aumentadas) foram aleatoriamente misturadas para evitar qualquer viés de classificação do sistema CAD. Os conjuntos de dados de treinamento e teste, mamografias, sofreram redimensionamento de tamanho para 448 x 448, pois, na base de dados possuem diferentes tamanhos. Além do redimensionamento, sofreram normalização na faixa de [0, 1] para obter um alto desempenho. Foi realizado uma correção da densidade periférica aplicando a técnica de *multi-threshold peripheral equalization* para remover o efeito da compressão da mama. Diferente de técnicas anteriores que aplicam o classificador em várias regiões da imagem, o YOLO utiliza uma única rede neural convolucional em toda a imagem. Divide a imagem de entrada em sub-regiões e prevê várias caixas delimitadoras com suas probabilidades de classe para cada região. YOLO é uma das técnicas de aprendizado profundo de última geração e é capaz de detectar e classificar objetos em imagens inteiras ao mesmo tempo. Diferentemente do *Region-based CNN* (R-CNN) tradicional, que requer muitas redes para todas as regiões extraídas, o YOLO utiliza mamografias inteiras para que as informações contextuais dos preditores e seus aspectos sejam completamente codificadas com uma única rede, tanto no tempo de treinamento quanto no de teste. O YOLO visualiza a imagem uma vez e não requer um pipeline complexo, é extremamente rápido e suas previsões são informadas pelo contexto global nos dados fazendo dele um sistema de detecção vantajoso sobre os demais.

Arquitetura YOLO é um sistema unificado que está habilitado para identificar possíveis ROIs e prever diretamente suas probabilidades de classe a partir de uma imagem inteira. Começa com a divisão de uma mamografia de entrada em uma grid de células $N \times N$ não sobrepostas. Cada célula da grid é responsável por detectar a massa potencial pertencente a essa célula. Conforme aplicado com sucesso em outros estudos, duas caixas delimitadoras com seus escores de confiança são utilizadas para representar cada célula da grid. A confiança é expressa como a probabilidade da massa existente multiplicada pela porcentagem de interseções sobre união (IOU) entre a verdade básica e as caixas previstas da seguinte maneira: $\text{Confiança} = \text{Probabilidade da Massa} \times \text{IOU}$. Além disso, a massa detectada é reconhecida como benigna ou maligna, dependendo da probabilidade condicional da classe $\text{Prob}(\text{Classe}_i | \text{massa})$ da célula correspondente: onde a pontuação de confiança interpreta a confiança do modelo para representar a massa envolvida na caixa prevista e também a precisão dessa massa; e esse índice de confiança se torna zero quando a célula da grid não contém nenhum objeto. A YOLO é treinada utilizando toda a imagem da mama com as informações de ROIs: Posição Central(x, y); Largura(W); Altura(h); e rótulo de cada classe das massas. Foram utilizados 24 camadas convolucionais com diferentes tamanhos de *kernel*, camadas de

pool máximo com tamanho 2×2 , funções de ativação e duas *fully-connected layers*. Os resultados dos testes utilizando *five-fold cross validation* mostram que o sistema CAD proposto detecta a localização da massa com uma precisão geral de 99,7%. O sistema também distingue entre lesões benignas e malignas com uma precisão geral de 97%. O sistema proposto ainda funciona em alguns casos desafiadores de câncer de mama, onde as massas existem sobre os músculos peitorais ou regiões densas.

3.3 A study of the suitability of autoencoders for preprocessing data in breast cancer experimentation

Macías-García et al. (2017) explora o uso de autocodificadores como técnica de pré-processamento, melhorando a qualidade dos dados experimentais usados para desenvolver assinaturas IHQ com valor prognóstico no câncer de mama. Para isso, o artigo testa autoencodificadores usando dados de pacientes com carcinoma invasivo de mama sem tipo especial (NST); Analisa sobrevida em relação à mortalidade, recorrência e comparação com os p-values de um conjunto de biomarcadores importantes no câncer de mama com os obtidos após o pré-processamento por outras técnicas (manual e binarização automática) usado como linha de base para a comparação; e por fim, estabelece uma estrutura estatística para validar os resultados e fornecer algumas ideias sobre como os autocodificadores podem melhorar os dados experimentais. O paper utiliza um banco de dados biomedico com 25 atributos para 222 pacientes com carcinoma invasivo de mama que não receberam tratamento neoadjuvante e foram acompanhados por um período de 5 anos. Foi comparado três métodos diferentes de pré-processamento: binarização manual; binarização automática; e binarização de autocodificadores. Na primeira, a responsabilidade de definir a ativação ou não de uma classe é exclusivamente humana. A segunda normaliza valores originais no intervalo $[0, 1]$, assumindo valores máximos e mínimos para cada característica. Após, discretizou os valores em 0 ou 1, técnica bem conhecida em machine learning. Na última, utiliza uma rede neural que deve decidir as características de dados mais importantes, atuando essencialmente como um mecanismo de *feature extraction*.

Após análise de sobrevivência utilizando pacote em R, tentou identificar fatores prognósticos independentes dos marcadores imuno-histoquímicos para outros resultados que não a sobrevida livre de doença, i.e., *recidive and exitus*. Para isso foi aplicado modelos *Cox Regression* após análise univariada com intervalo de confiança de 95%. Também foi avaliado as inclinações em regressões lineares generalizadas da escala *Schoenfeld* residual em função do tempo (TREC e TMORT). Uma inclinação non-zero a respeito de um predictor é uma indicação de uma violação da suposição de risco proporcional que é necessária para garantir a validade das conclusões alcançadas por uma regressão de Cox. Foi utilizado *Cox Regression*

no R que retornou p-values associados com cada preditor. Cada p-value mostrou a significância de uma covariável específica em relação a uma variável dependente (REC, MORT). Para teste das 3 opções comparadas, o foco foram os indicadores mais importantes para câncer de mama: ER, PR, Ki-67 e HER2.

Para evitar o *overfitting* devido a parametrização feita com todos os dados de paciente, um procedimento de *hold-out* foi feito para avaliar a validade das conclusões. Com isso, o conjunto de dados foi dividido em dois subconjuntos aleatórios: treinamento e teste. O primeiro foi usado para criar automaticamente as técnicas de pré-processamento e para a parametrização do *autoencoder*; o segundo, após a parametrização do *autoencoder* foi aplicado cada técnica de pré-processamento para transformar o conjunto de teste. E por fim, ainda no subconjunto de teste, foi aplicado uma análise de sobrevivência ao conjunto de testes transformado e acumulando os p-values correspondentes. Para cada técnica de pré-processamento comparada o procedimento completo foi repetido dez vezes para evitar viés devido à seleção aleatória inicial e foi calculada a média dos p-values. De acordo com os p-values encontrados ER, PR, Ki-67 e HER2 em relação a MORT e REC na regressão de Cox da binarização automática após a *autoencoding* seria melhor qualitativamente que as outras opções, mas não quantitativamente.

3.4 *Deep Convolutional Neural Networks Enable Discrimination of Heterogeneous Digital Pathology Images*

Khosravi et al. (2018) propõe uma arquitetura básica de RNC e várias arquiteturas são comparadas em um pipeline de código aberto aplicável a um grande número de imagens histopatológicas. Mesmo com a heterogeneidade do tumor, o pipeline de classificação alcançou precisão de 100%, 92%, 95% e 69% para discriminação de vários tecidos, subtipos, biomarcadores e escores de coloração IHQ para câncer de pulmão, bexiga e mama. A RNC básica usou dois algoritmos do estado da arte: *Inception* e *ResNet*. treinou a rede a partir do zero, fez ajuste fino dos parâmetros usando duas versões pré-treinadas das arquiteturas *google's Inception*, e treinamento da última camada usando *Inception*. O conjunto de dados utilizado foi TMAD e TCGA com imagens inteiras de tecidos corados por HE e IHQ, biomarcadores para neoplasia do pulmão, mama e bexiga. As amostras de tecido foram pontuadas por um patologista treinado usando um sistema de pontuação discreta. As imagens inteiras foram usadas diretamente como entrada e divididas em diferentes classes com base nos objetivos de classificação. A proporção 70:15:15 para treinamento, validação e teste foi a que melhor resultados trouxe. O objetivo da arquitetura é discriminar diferentes tipos de câncer, subtipos, marcadores imuno-histoquímicos e escores de coloração de marcadores de imagens histopatológicas. Para avaliar o desempenho dos algoritmos foram utilizados medidas

estatísticas: *AUC, Precision, Recall, Cohen's kappa, Jaccard coefficient e Log-loss*.

O processo de treinamento foi parado quando a precisão da validação convergia ao máximo. Esse ponto de parada foi considerado como o *optimal step number* para a arquitetura RNC e o conjunto de dados testado. Após a classificação final das imagens no conjunto de testes foi realizado novo treinamento e validação com o novo número ideal. A precisão de diferentes estratégias de treino encontradas usando *Inception V1* foram: ajuste fino dos parâmetros para todas as camadas e treinamento do zero ficaram com 90% e 86% respectivamente e; treinamento da última camada ficou com aproximadamente 76%. O melhor desempenho foi usando uma rede pré-treinada e ajustando os parâmetros para todas as camadas da rede, estando de acordo com os resultados de estudos anteriores. A separação de vários marcadores IHQ da bexiga resulta em precisão de 71,5% a 99% para o ajuste fino do RNC básica e do *Inception V1*, respectivamente. A aplicação dos algoritmos mencionados nos marcadores imunohistoquímicos da mama leva a uma precisão de 79,2% e 90%, RNC básica e *Inception V3* respectivamente. Em dois casos de duzentos, o classificador *Inception V1* classificou erroneamente os biomarcadores S0084 e S100P com GATA3 e S0084, respectivamente, atingindo uma precisão de 99% para o caso de câncer de bexiga. O resultado *Inception V1* (90%) para discriminação de biomarcadores de mama revelou que todas as contradições de 10% ocorreram entre CK17 e CK5 / 6 devido à alta similaridade entre eles. Os experimentos foram projetados para investigar se manter a cor do plano de fundo pode ter o potencial de introduzir certos vieses inerentes nos conjuntos de dados e afetar o resultado da discriminação de vários marcadores. A precisão de interpretação de imagens médicas, pelos médicos, é reduzida devido a subjetividade, a grande variações através das interpretações, e a exaustão. Mesmo com viés de erro humano para a classificação de bexiga afetada para o câncer de mama, o algoritmo RNC com arquitetura básica pode discriminar as imagens positivas (escores 1, 2 e 3) e negativas (score 0) com 94% de precisão. A aplicação do *Inception V3* (treinamento da última camada) indicou 96% de precisão para o mesmo conjunto. RNC básica é capaz de detectar diferentes tecidos, subtipos e marcadores relacionados ao câncer com precisão altamente confiável, que depende do conteúdo do conjunto de dados, do tamanho do conjunto de dados e do algoritmo selecionado. A arquitetura RNC apresentada fornece uma estrutura que pode ser facilmente implementada para uma ampla variedade de aplicações, incluindo classificação imunohistoquímica e detecção de biomarcadores tumorais.

3.5 Detection of breast cancer on digital histopathology images: Present status and future possibilities

Aswathy e Jagannath (2017) analisa e resume as aplicações das técnicas de processamento digital de imagens em imagens histopatológicas para a detecção de câncer de mama e discute

suas possibilidades futuras do sistema de diagnóstico assistido por computador. O paper segue um roteiro que inicia escrevendo sobre os tipos de tingimento e modalidades de imagem na histopatologia; destaca vários métodos usados para análise de imagem histopatológica; os diferentes parâmetros usados para avaliação de um classificador; e discute e compara diferentes algoritmos usados para detecção, segmentação e classificação de núcleos.

3.6 Deep Multi-instance Networks with Sparse Label Assignment for Whole Mammogram Classification

Zhu et al. (2017) traz uma proposta de explorar o contrário da classificação tradicional que requer *Bounding Box For Detection ou Mask Ground Truth For Segmentation* necessitando de trabalho manual para treino. Portanto, o paper utilizou *Deep Multi-instance Networks* para classificação em massa com base em mamografias completas sem utilizar das regiões de interesse. Para isso, usou três métodos para construção da *Deep Multi-instance Network: Max Pooling, Label Assignment e Sparsity*. Uma das motivações é que *Deep Multi-instance Networks* com DL é um tópico emergente. Para explorar a *Deep Multi-instance Networks* como evolução da classificação tradicional, o paper assume uma *Label Assignment* que diferente da *Max Pooling* considera todos os fragmentos para classificação. Adiciona prioridades relacionadas as tarefas na suposição de várias instâncias e explora mais fragmentos para treinamento, e.x., escassez de massa na mamografia completa. Depois de classificar todas as probabilidades malignas para todas as instâncias, os primeiros fragmentos devem ser consistentes com o label da mamografia inteira, enquanto os demais devem ser negativos.

Tabela 5 – Comparativo dos trabalhos relacionados

Artigo	Rede	Imagem	Dataset	Tamanho Dataset	ROI	Coloração	Método
(KIM et al., 2019)	Decision Jungle	-	Privado	Pequeno	-	-	U teste, qui-quadrado, AUC e ROC
(MACÍAS-GARCÍA et al., 2017)	MLP	-	Privado	Pequeno	-	-	Cox Regression
(AL-MASNI et al., 2018)	Yolo	Inteira	Público	Grande	Sim	-	Five-fold cross validation
(KHOSRAVI et al., 2018)	Inception e ResNet	Inteira	Público	Grande	Sim	H/E e IHC	AUC, Cohen's kappa, Jaccard coefficient
(ZHU et al., 2017)	AlexNet	Inteira	Privado	Pequeno	Não	-	Five-fold cross validation
(ASWATHY; JAGANNATH, 2017)	GCNN	Ambos	Público	Grande	Sim	H/E e IHC	Cross validation, AUC e coeficiente de correlação Matthew's

Fonte: Desenvolvido pelo autor.

O trabalho visa explorar, através de imagens de exame de imuno-histoquímica a detecção e classificação da alta expressividade da membrana celular. Para isso faz uso de técnicas de pré-processamento como: *data augmentation*, filtros *kernel*, operações morfológicas. Propõe o uso de dois algoritmos para extração de recursos, compara seus resultados utilizando uma support vector machine e também com outra rede neural artificial profunda. As abordagens de rede neural artificial deste trabalho utilizam as curvas de aprendizado e a matriz de confusão como base para a comparação.

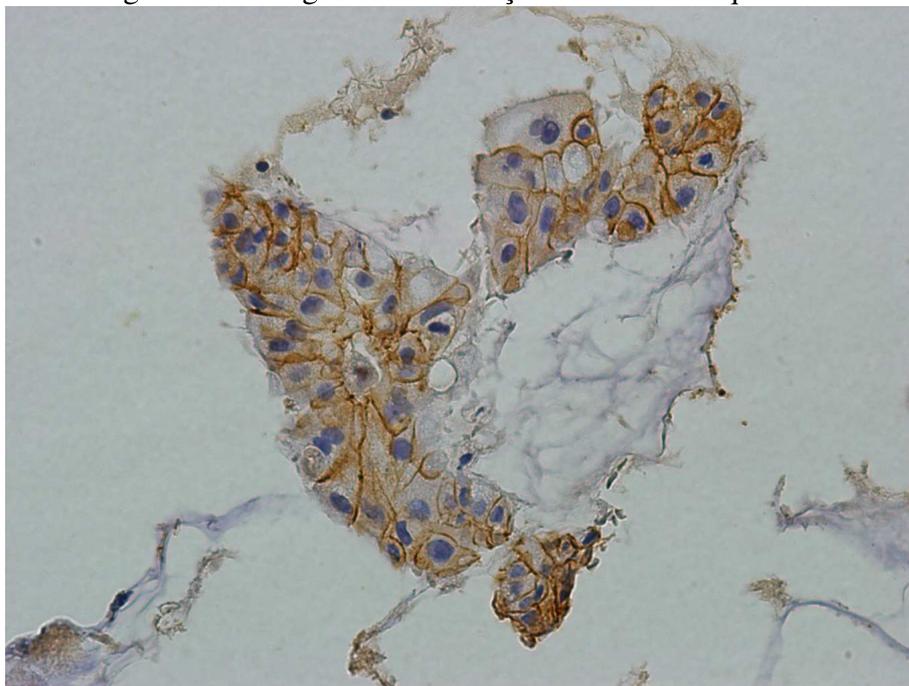
4 MODELO PROPOSTO

Neste capítulo apresentamos os materiais e métodos utilizados pelos dois modelos, independentes e distintos, que são de rede neural artificial profunda e rede neural tradicional utilizando algoritmos baseados em textura para extração de recursos. Ambos possuem como objetivo a detecção e classificação da hiperexpressão da proteína envolvida no transporte epitélio-mesenquimal a partir de imagens de exames de IHQ utilizando regiões de interesse e tentam reproduzir métodos quantitativos de detecção de marcadores.

4.1 Material e Métodos

A ideia central da abordagem proposta é reproduzir a rotina diária do profissional de patologia no que diz respeito ao processo de quantificação da hiperexpressão, através da membrana celular, auxiliando o diagnóstico. Portanto, o foco da detecção e classificação da proteína envolvida no transporte epitélio-mesenquimal será a membrana celular.

Figura 11 – Imagem com as reações imuno-histoquímica



As reações IHQ foram realizadas no Laboratório de Pesquisa em Patologia da UFCSPA, conforme o procedimento operacional padrão do laboratório. Na realização do estudo imuno-histoquímico utilizou-se espécime tumoral fixado em formalina 10%, incluso em parafina. Os blocos foram seccionados em espessura de $3\mu m$, desparafinados e reidratados. Para ambas as proteínas a recuperação antigênica foi realizada com Tris/EDTA (pH 9,0), aquecido, 30 minutos, em banho-maria à temperatura de $92^{\circ}C$. Em seguida, as lâminas foram

lavadas em água destilada. A atividade da peroxidase endógena foi bloqueada, utilizando-se três banhos de 10 min em peróxido de hidrogênio (H_2O_2) 30V, a 5%, em metanol, posterior repetiu-se a lavagem anterior e mais duas em PBS, durante 5 minutos por agitação. O bloqueio das proteínas inespecíficas foi realizado com BSA (albumina de soro bovino fração V) 1% por 1 hora. Utilizou-se anticorpo monoclonal mouse para detecção das proteínas, nas seguintes titulações: E-caderina 1/300 (G-10 Santa Cruz Biotechnology) e N-caderina 1/100 (6G-11 DAKO), incubando-se durante toda a noite, overnight, à temperatura de 4°C. Para os anticorpos secundários e terciários, utilizou-se o Kit Picture MAX Polymer, incubando-se por 40min (cada) à temperatura ambiente. Na incubação do controle interno negativo, foi utilizado uma amostra positiva do mesmo tecido, com os mesmos anticorpos, exceto o primário, que foi substituído por BSA 1%. Para controle positivo externo, foi utilizado tonsila palatina. A ligação antígeno-anticorpo foi visualizada com o cromógeno DAB (diaminobenzidina). A contra coloração foi feita com hematoxilina de Harris (Merck); as lâminas foram desidratadas e montadas com resina sintética.

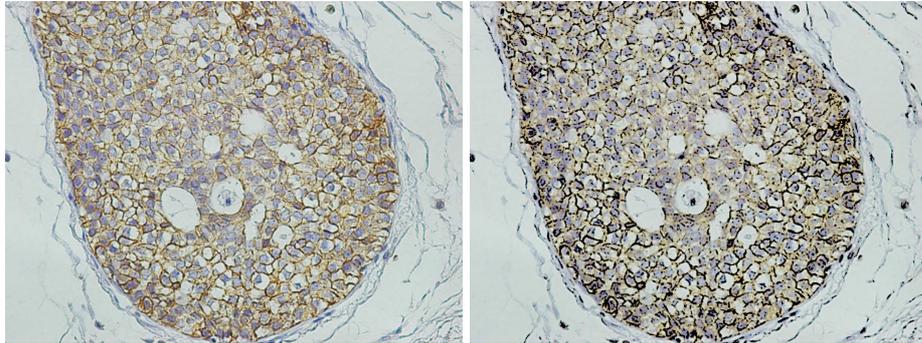
Posterior ao procedimento operacional padrão do laboratório para a realização das reações IHQ, as lâminas de 143 pacientes do sexo feminino, diagnosticadas positivamente para o carcinoma da mama, foram digitalizadas em aumento de 400 X, utilizando o software (DP2-BSW) da câmera (Olympus DP72) acoplada ao microscópio (Olympus BX51), conectado ao computador. Áreas com hotspots IHQ foram registradas de forma não compactada Tiff (Tagged Image Format), com resolução de 1360 x 1024 pixels e 24 bits, no modelo de cor RGB (*Red, Green, Blue*), e arquivos de tamanho de 5,3 megabyte. Visando obter dados quantitativos sobre a expressão dos anticorpos foi utilizado o método de deconvolução de cor. A Figura 11 mostra uma imagem de estudo celular que sofreu o processo de reação química pela imuno-histoquímica.

Assim como (GARCIA et al., 2017) sugere uma melhora no conjunto de dados e (ASWATHY; JAGANNATH, 2017) destaca o uso de operações morfológicas, a abordagem proposta também utiliza dessas técnicas na fase de pré-processamento. A essência do processamento de imagem é usar as diferentes propriedades de uma imagem, como cor, correlações entre diferentes pixels, posição do objeto e outros detalhes finos para extrair informações significativas, como bordas, objetos e contornos, que são formalmente chamados de recursos de imagem. Esses recursos podem então ser usados em diferentes aplicações, para direcionar imagens médicas, sistemas de segurança, serviços de mídia social e carros autônomos, entre várias outras aplicações comerciais. A etapa de pré-processamento é talvez a mais importante na aplicação de modelos de RNAs.

Portanto, é sempre necessário realizar uma ou mais etapas de pré-processamento para o sucesso do modelo da RNA. Em seguida, o conjunto de dados utilizado neste trabalho passou

por alguns processos de filtragem e operações morfológicas. Para remover o ruído da imagem, duas operações morfológicas foram aplicadas: erosão e dilatação. No processo de filtragem, desfocagem gaussiana e *threshold image* foram usados. O desfoque gaussiano usa uma média ponderada, onde os pixels na vizinhança que estão mais próximos do pixel central contribuem com mais "peso" para a média. O *threshold image* permite que separe os objetos importantes do fundo. A Figura 12 mostra uma imagem de exame IHQ antes e após etapa de pré-processamento.

Figura 12 – Imagem com as reações IHQ antes e após pré-processamento



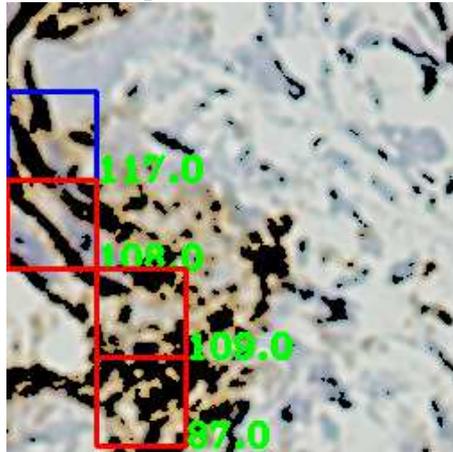
Fonte: Elaborado pelo autor.

A fim de atenuar possíveis problemas de *overfitting* decorrente do pequeno número de imagens e melhor generalização do modelo foram geradas novas imagens a partir da original, rotacionando a mesma em 90, 180 e 270 graus. Ainda seguindo no processo de *data augmentation* para cada nova imagem gerada foi feito o *crop*, dividindo em novas imagens de 256 x 256 pixels gerando 12.421 imagens.

Diferente de Zhu et al. (2017) o trabalho usa regiões de interesse para classificar a hiperexpressão da membrana celular. As novas imagens de 256 x 256 pixels foram usadas para a detecção de regiões de interesse (Roi) ou *key points*. Um intervalo de valores [50, 115, 125, 145] foi utilizado para encontrar a média em pedaços de 50 x 50 pixels. Essas áreas representam o grau de expressão e para cada uma foi dado as seguintes rótulos: nenhuma (branco), alta (vermelho), média (azul) e baixa (verde). Como o objeto a ser detectado não possui forma definida, a escolha da textura foi a melhor forma para encontrar os *key points*. A Figura 13 abaixo mostra os key points de 50 x 50 pixels detectados com suas respectivas classificações na imagem 256 x 256 pixels.

A partir das imagens de 256 x 256 pixels foram gerados 36.515 patches de 50 x 50 pixels com seus rótulos: alta, média, baixa e nenhuma. Foram utilizados como conjunto de dados para treinamento e teste 36.515 patches com seus labels: nenhuma, baixa, média e alta, como mostra a Figura 14.

Figura 13 – Key points de 50 x 50 pixels detectados em imagens de 256 x 256 pixel



Fonte: Elaborado pelo autor.

Figura 14 – Patches de 50 x 50 pixels classificados como alta imunexpressão



Fonte: Elaborado pelo autor.

Diferentemente das RNC, algumas RNA precisam receber como entrada um vetor de recursos pois não possuem as etapas de convolução para geração desses recursos. Portanto essa é uma das diferenças das DL para as RNA tradicionais. Dois algoritmos foram usados para extrair um vetor de recursos baseado em textura. O vetor de recursos gerado juntamente com seu respectivo rótulo foi usado para treinar e testar o modelo. Os dois algoritmos baseados em textura usados são: *Local Binary Pattern* e *haralick*, apesar de tratarem textura possuem saídas diferentes.

O recurso de imagem mais fácil e intuitivo para a maioria das aplicações em análise de imagens são estatísticas de primeira ordem calculadas a partir de histogramas dos valores de nível de cinza em imagens, como sua média, variância, assimetria e curtose. Esses recursos envolvem os valores de *pixels* individuais, mas ignoram a interação espacial entre os *pixels*. Os recursos do histograma não refletem objetos ou padrões na imagem, apenas a distribuição dos níveis de cinza. Essa incapacidade torna as estatísticas de primeira ordem uma ferramenta contundente para quantificar mudanças nas imagens ou qualquer mudança na distribuição espacial dos valores de cinza.

O LBP lê a imagem, procura *pixels* adjacentes e gera um *pixel* de saída semelhante. É muito semelhante ao mecanismo de *kernels* que são aplicados a uma imagem e geram um resultado. A diferença é que a saída do algoritmo LBP não é um vetor de recursos, mas um conjunto numérico que não representa um vetor de recursos. Como resultado, é necessário converter a saída em um histograma. Assim, é possível ter um vetor de recursos ao usar o algoritmo LBP.

O algoritmo LBP é definido como um descritor de textura robusta (AGRAVAT et al., 2018). No LBP, os valores de *pixel* das imagens segmentadas são comparados ao valor de *pixel* central para calcular um limite. Dependendo do limite, os recursos são extraídos. Este método provou ser uma ferramenta poderosa para extrair recursos de textura. Neste método, a janela de destino é dividida em várias células e o *pixel* é comparado com os *pixels* vizinhos. Se o valor do pixel for maior ou igual ao *pixel* central, as células são rotuladas como "0" ou "1". Depois de rotular os valores de *pixel* com os códigos LBP correspondentes, dependendo dos valores de *pixel*, o histograma é calculado. Após adicionar o histograma de todas as células, o vetor característico é obtido. O histograma pode ser definido como:

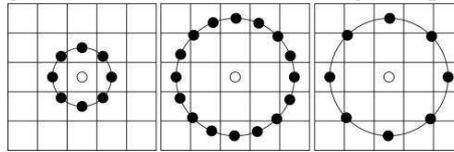
$$H_i = \sum xyI(f(x, y)) = i \quad (4.1)$$

$i = 0; \dots (n - 1)$ e n é um rótulo diferente produzido pelo código LBP. $I(A) = 0$, quando A é falso e $I(A) = 1$, quando A é verdadeiro. Este método é muito popular para abordagens de detecção de rosto e reconhecimento de padrões. O operador LBP transforma uma imagem em uma matriz ou imagem de rótulos inteiros que descrevem a aparência em pequena escala da imagem.

O operador também pode ser estendido para usar vizinhanças de tamanhos diferentes (Ojala; Pietikainen; Maenpaa, 2002). O uso de vizinhanças circulares e a interpolação bilinear dos valores de *pixel* permite qualquer raio e número de *pixels* na vizinhança. Para vizinhanças, usaremos a notação (P, R), que significa P pontos de amostragem em um círculo de raio de R, como mostra a Figura 15.

Uma extensão útil para o operador original é o chamado padrão uniforme (Heikkila; Pietikainen, 2006), que pode ser usado para reduzir o comprimento do vetor de características e implementar um descritor invariante de rotação simples. Essa ideia é motivada pelo fato de que alguns padrões binários ocorrem mais comumente em imagens de textura do que outros. Um LBP é denominado uniforme se o padrão binário contiver no máximo duas transições 0-1 ou 1-0. Por exemplo, 00010000 (duas transições) é um padrão uniforme, 01010100 (seis

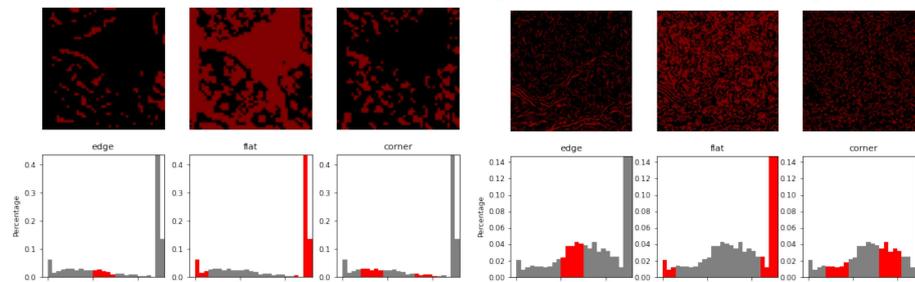
Figura 15 – LBP na vizinhança do pixel



Fonte: Elaborado pelo autor.

transições) não. Para o cálculo do histograma de LBP, o histograma é composto de um compartimento separado para cada padrão uniforme, e todos os padrões não uniformes são atribuídos a um único compartimento, como mostra a Figura 16. Usando padrões uniformes, o comprimento do vetor de característica para um célula única é reduzida de 256 para 59.

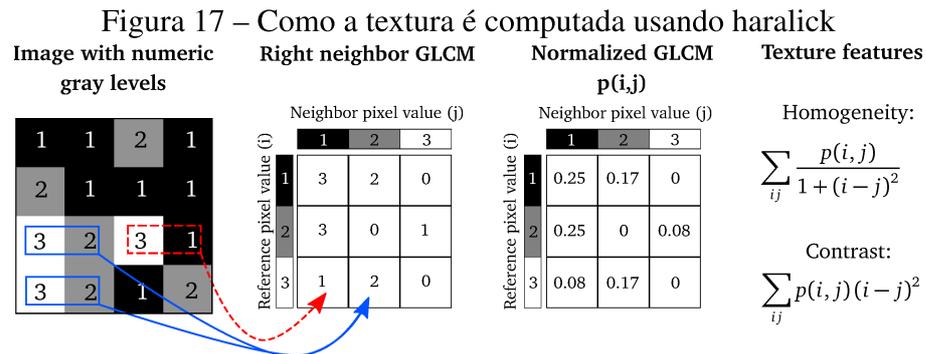
Figura 16 – Histograma de uma imagem com e sem imunoe expressão



Fonte: Elaborado pelo autor.

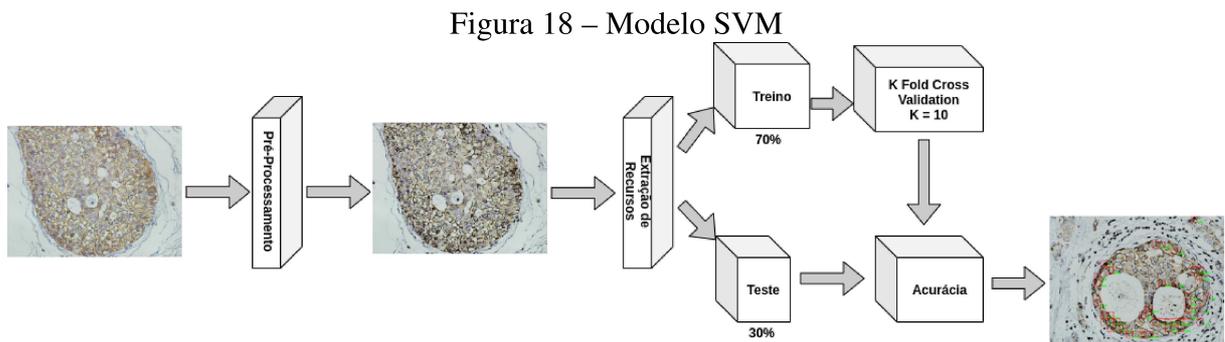
Os recursos de *Haralick* também descrevem a textura de uma imagem, mas, ao contrário do algoritmo LBP, possui um vetor de recursos como saída. Esta textura refere-se à aparência, consistência ou "sensação" de uma superfície. Exemplos de textura incluem "áspera" versus "macia". Nos últimos anos, tem havido um rápido aumento na aplicação de recursos *Haralick* na análise de imagens médicas, por ex. na análise de ultrassom e imagens de ressonância magnética do fígado, do coração, mamografia de raios-X, imagens de ressonância magnética no estudo de câncer de mama, câncer de próstata e câncer de cérebro (LÖFSTEDT, 2019). Os recursos *Haralick* são calculados usando a Matriz de Co-ocorrência de Nível de Cinza (GLCM). Essa matriz caracteriza a textura, registrando a frequência com que pares de *pixels* adjacentes com valores específicos ocorrem em uma imagem. O algoritmo extrai características de textura em 4 direções como um *kernel*, então calcula a média de cada direção, veja na Figura 17 uma imagem 4 x 4 onde três níveis de cinza são representados por valores numéricos de 1 a 3.

Ao criar um GLCM a partir de uma imagem ou ROI, a profundidade de bits da imagem, ou seja, o número de níveis de cinza, pode ser reduzida em um processo denominado quantização. O número de níveis de cinza na imagem quantizada determina o tamanho do GLCM e afetará os valores do recurso de textura. Ao aumentar o número de níveis de cinza, N , as somas nas



Fonte: (LÖFSTEDT, 2019).

Tabelas 1 e 2 mudarão. Além disso, os valores do elemento, $p(i, j)$, do GLCM normalizado diminuirão conforme o número de níveis de cinza aumenta, uma vez que a soma de todos os elementos é a unidade. Estas duas propriedades do GLCM e das características de textura *Haralick* significam que os valores das características de textura dependem do número de níveis de cinza na imagem quantizada (COELHO, 2013).



Fonte: Elaborado pelo autor.

A Figura 18 apresenta o modelo de rede neural artificial SVM que utiliza dois vetores de recursos, LBP e *Haralick*. Onde o conjunto de dados previamente rotulado, 50 x 50 pixels, foi dividido da seguinte forma: 70 % para treinamento e 30 % para teste. Ao final, 12 imagens inteiras foram utilizadas para validação, ou seja, imagens de 1360 x 1024 pixels.

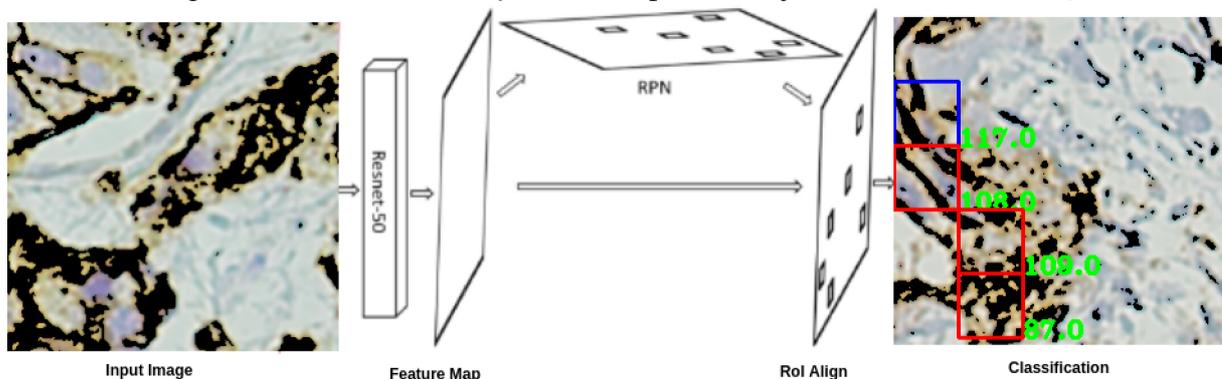
A detecção de objetos é uma tarefa desafiadora de visão computacional que envolve prever onde os objetos estão na imagem e quais tipos de objetos foram detectados. Assim, mesmo que o foco do trabalho seja a detecção a partir da textura, um outro modelo de RNA foi utilizado.

He et al. (2017) apresentam um *framework* simples, flexível e genérico para segmentação de instância de objetos. A abordagem detecta objetos em uma imagem de forma eficiente e, ao mesmo tempo, gera uma máscara de segmentação de alta qualidade para cada instância. O método chamado de Mask R-CNN, estende o Faster R-CNN, adiciona a predição de uma

máscara de segmentação em cada RoI, em paralelo destaca a existência de uma caixa delimitadora para classificação e regressão. Como a arquitetura é utilizada neste trabalho é apresentado na Figura 19 e compreende as seguintes etapas: Imagem de entrada, ResNet-50, extração dos mapas, RPN, *RoI Pooling* e classificação. O mapa de características é extraído da imagem de entrada usando a ResNet-50 seguida pela geração das âncoras através da RPN e a detecção final da hiperexpressão na rede de classificação. A Mask R-CNN tem a vantagem de ser um *framework* de fácil generalização a muitas tarefas, portanto, esse foi um dos motivos pelo qual foi utilizada neste trabalho.

Uma Mask R-CNN é uma das abordagens mais avançadas para tarefas de reconhecimento de objetos. Abdulla (2017) fornece uma Mask R-CNN, uma biblioteca que permite desenvolver e treinar modelos Mask R-CNN Keras para tarefas de detecção de objetos. Nesta abordagem foi permitido um treinamento das imagens IHQ por meio de TL com modelos pré-treinados. Isso foi feito eliminando algumas das últimas camadas existentes e utilizando o arquivo de pesos da rede pré-treinada por (ABDULLA, 2017).

Figura 19 – Mask R-CNN *framework* para o conjunto de dados de IHQ



Fonte: Elaborado pelo autor.

O desempenho de um modelo para uma tarefa de reconhecimento de objeto é frequentemente avaliado usando a precisão absoluta média (do inglês mAP).

Como em (KHOSRAVI et al., 2018) estamos prevendo caixas delimitadoras para que possamos determinar se uma previsão de caixa delimitadora é boa ou não com base em quão bem as caixas delimitadoras previstas e reais se sobrepõem. Isso pode ser calculado dividindo a área da sobreposição pela área total de ambas as caixas delimitadoras ou a interseção dividida pela união, conhecida como IoU, como mostra a Equação 4.2.

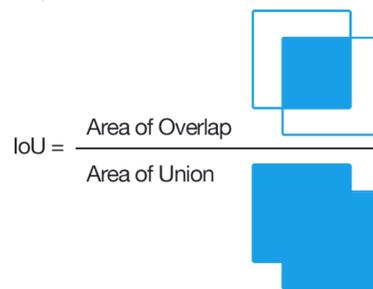
Essa métrica de avaliação é frequentemente usada para segmentação de imagens, pois é mais estruturada. O IoU é uma métrica de avaliação por classe, que calcula o número de pixels

na interseção entre os mapas de segmentação de verdade prevista e de base para uma determinada classe, dividido pelo número de pixels na união entre esses dois mapas de segmentação, também para o dado classe como mostram a Figura 20.

$$IoU(P(class), GT(class)) = \frac{[P(class) \cap GT(class)]}{[P(class) \cup GT(class)]} \quad (4.2)$$

onde P é o mapa de segmentação previsto e GT é o mapa de segmentação da verdade básica. $P(class)$ é então a máscara binária indicando se cada pixel é previsto como classe ou não. Em geral, uma previsão de caixa delimitadora perfeita terá um IoU de 1.

Figura 20 – Visualização IoU



Fonte: (LONG; SHELHAMER; DARRELL, 2015).

Neste trabalho o valor assumido para uma previsão positiva de uma caixa delimitadora (do inglês *bounding box*) se o IoU for maior que 0,7, portanto, eles se sobrepõem em 70% ou mais.

O *precision* se refere à porcentagem das caixas delimitadoras corretamente previstas ($IoU > 0,7$) de todas as caixas delimitadoras previstas. O *recall* é a porcentagem das caixas delimitadoras corretamente previstas ($IoU > 0,7$) de todos os objetos na imagem.

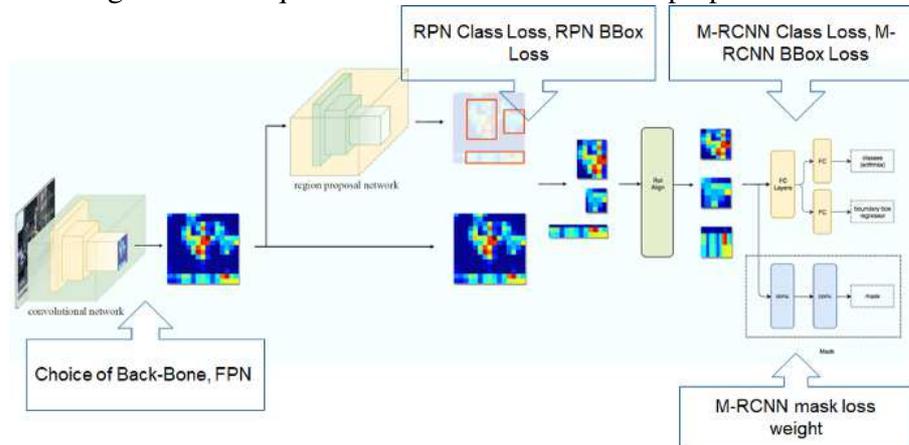
À medida que fazemos mais previsões, a porcentagem de *recall* aumentará, mas o *precision* cairá conforme começamos a fazer previsões falsas positivas. O $recall(x)$ pode ser plotada contra o $precision(y)$ para cada número de previsões para criar uma curva ou linha. Podemos maximizar o valor de cada ponto nesta linha e calcular o valor médio da precisão ou AP para cada valor de *recall*.

Um dos problemas da Faster R-CNN é não ser projetada para alinhamento *pixel-to-pixel* entre as entradas e saídas da rede. Portanto, para resolver esse problema (HE et al., 2017) propõe uma simples camada livre de quantização, chamada RoIAlign que preserva fielmente as localizações espaciais exatas. Apesar de ser uma alteração aparentemente pequena, o RoIAlign tem um grande impacto: melhora a precisão da máscara em cerca de 10% a 50%

A Figura 21 ilustra a arquitetura Mask R-CNN. Uma descrição de cada passo da arquitetura Mask R-CNN pode ser observado abaixo:

1. A imagem é passada por uma rede convolucional que em nosso caso é uma ResNet50;
2. A saída da primeira rede convolucional é passada adiante para uma RPN que cria diferentes caixas de âncoras (RoIs) baseado na presença de qualquer objeto a ser detectado;
3. Essas caixas de âncoras são enviadas para o RoIAlign (uma das características chaves da Mask R-CNN por manter orientação espacial) que converte essas RoIs para o mesmo tamanho requerido em processamento posterior;
4. Esta saída é enviada para as camadas FCNs que irão gerar o resultado da classe do objeto na região específica e a localização da *bounding box* do objeto;
5. A saída do RoIAlign é enviada paralelamente para a Conv Net com a finalidade de gerar a máscara (do inglês Mask) de *pixels* do objeto.

Figura 21 – Arquitetura Mask R-CNN com hiperparâmetros



Fonte: (ABDULLA, 2017)

Alguns hiperparâmetros específicos para Mask R-CNN:

1. *Back Bone*: é a arquitetura Conv Net que deve ser utilizada na primeira etapa do Mask R-CNN. As opções disponíveis para escolha de Backbones incluem ResNet50, ResNet101 e ResNext 101;
2. *Train ROIs Per Image*: Este é o número máximo de RoIs que a RPN irá gerar para a imagem, que será posteriormente processada para classificação e mascaramento na próxima etapa;
3. *Max GT Instances*: Este é o número máximo de instâncias que podem ser detectadas em uma imagem;

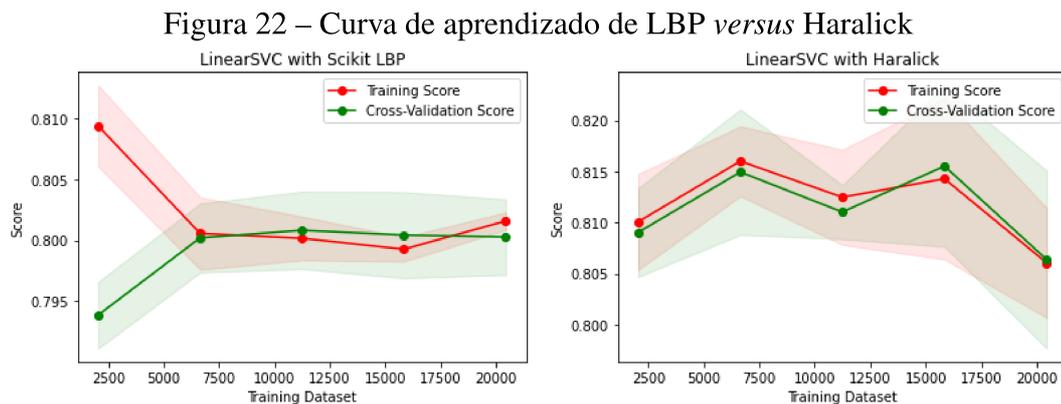
4. *Detection Min Confidence*: Este é o limite do nível de confiança, além do qual ocorrerá a classificação de uma instância;
5. *Image Min Dim e Image Max Dim*: O tamanho da imagem é controlado por essas configurações. As configurações padrão redimensionam as imagens para quadrados de tamanho 1024x1024;
6. *Loss weights*: Os hiperparâmetros de peso de perda correspondem ao peso que o modelo deve atribuir a cada uma de suas etapas
 - (a) Rpn class loss: Corresponde à perda que se atribui à classificação indevida das âncoras (presença / ausência de qualquer objeto) pela RPN;
 - (b) Rpn bbox loss: Isso corresponde à precisão de localização do RPN;
 - (c) Mrcnn class loss: Corresponde à perda que é atribuída à classificação inadequada do objeto que está presente na região proposta;
 - (d) Mrcnn bbox loss: Esta é a perda, atribuída na localização da caixa delimitadora da classe identificada;
 - (e) Mrcnn mask loss: Corresponde às máscaras criadas nos objetos identificados.

Da técnica de *data augmentation* da abordagem anterior foram usadas, na arquitetura Mask R-CNN, 162 imagens de IHQ com tamanho de 256 x 256 *pixels*. Onde para treino foram utilizadas 137 imagens de 256 x 256 *pixels*, teste 25 imagens de 256 x 256 *pixels* e 12 imagens inteiras de 1360 x 1024 *pixels* para validação. Cada imagem de treino e teste possui um arquivo XML com as segmentações encontradas pelo cálculo da média na etapa de pré-processamento.

Para treino da Mask R-CNN foi utilizado como *backbone* uma ResNet-50 com arquivo de peso de uma rede pré-treinada (*mscoco*) e devido ao uso da técnica de TL, as últimas quatro camadas da rede pré-treinada foram excluídas. Dessa forma, o conjunto de dados de IHQ foi treinado pela arquitetura Mask R-CNN utilizando arquivo de peso existente e 25 épocas de treino com alguns parâmetros: *batch size*, *momentum*, *learning rate* e *weight decay*, e seus respectivos valores 1, 0,9, 0,003, 0,0005.

5 RESULTADOS E DISCUSSÕES

Nesta seção, apresentamos, inicialmente, uma comparação de resultados entre o LBP e o *haralick*. Com o Linear SVC, obtemos aproximadamente 80 % e 81 % de precisão para o classificador utilizando os dois algoritmos de textura para extração de características.

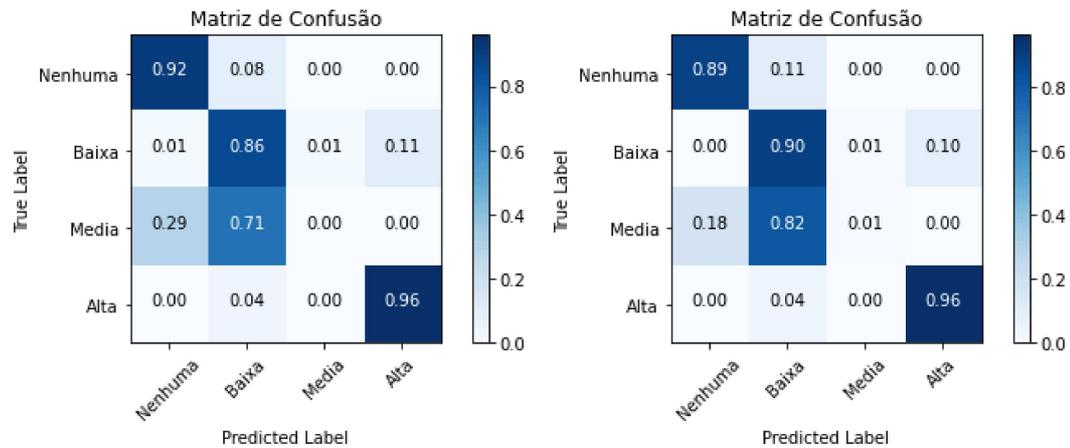


Fonte: Elaborado pelo autor.

Apesar da matriz de confusão apresentar números expressivos de verdadeiramente positivo e falso negativo ou positivo como mostram as Figuras 23 e 24, a curva de aprendizado, apresentada na Figura 22, possui a linha de *train score* muito próxima da linha de *cross-validation score* sugerindo que o modelo pode estar sofrendo *overfitting*. Entretanto, a fim de atenuar ao máximo esse fenômeno, o modelo foi treinado e testado, usando um dataset de 36.515 patches de imagens de 50 x 50 pixels, onde 10.955 imagens foram utilizadas para teste. Juntamente com a curva de aprendizado, é apresentada uma matriz confusão para os resultados produzidos pela SVM, a fim de verificar que não encontramos muitos falsos positivos.

A matriz de confusão normalizada dos modelos apresentam valores significativos de falso negativo e falso positivo mas vale ressaltar os valores encontrados para a classe "Média". Esta classe em ambos modelos possui altos valores de falso positivo. Ou seja, altos valores de falso positivo para a classe "Baixa". Se levarmos em consideração o número de *support* da classe "Média" apresentado na Tabela 6, fica evidente que essa situação contribui para uma queda de precisão em ambos os modelos.

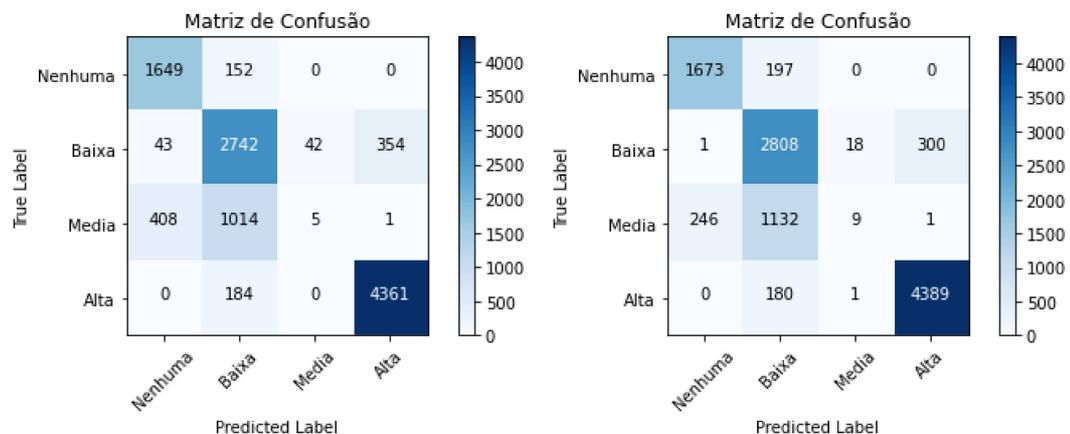
Para medir os resultados, foi aplicado uma matriz de confusão para obter alguns *scores* relevantes (*accuracy*, *error rate* ($1 - accuracy$), *confidence interval*, *precision*, *recall*, *specificity*, *log loss*, and *F1 score*). Em relação a esses indicadores, destacamos a *accuracy*, *log loss*, *F1 score*, *precision* e *recall* nesta pesquisa, apresentados a seguir.

Figura 23 – Matriz de Confusão Normalizada de LBP *versus* Haralick

Fonte: Elaborado pelo autor.

A estimativa de precisão é a soma das classificações corretas em todas as iterações dividida pelo número total de tuplas nos dados iniciais (HAN; PEI; KAMBER, 2011). É medido através de uma matriz de confusão onde quatro possíveis casos de predição são considerados: verdadeiro positivo (TP), verdadeiro negativo (TN), falso positivo (FP) e falso negativo (FN) (MONTAZERI et al., 2016).

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (5.1)$$

Figura 24 – Matriz de Confusão de LBP *versus* Haralick

Fonte: Elaborado pelo autor.

A *F1 score* é baseada em uma equação que usa a média de dois outros indicadores (*precision* e *recall*), interpretados como uma média ponderada dos dois (ASRI et al., 2016).

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (5.2)$$

O uso dos algoritmos de extração de textura pela SVM tem considerável baixo valor para a média macro devido a classe "Média" possuir valores muito pequenos de precisão, *recall* e *f1 score* evidenciando o alto número de falsos positivos. Por outro lado, a média ponderada que representa o total de verdadeiros positivos sobre o total de números de objetos em todas as classe, possui valores satisfatórios. Também na Tabela 6, é identificado uma situação de difícil comparação para os dois modelos, onde a classe "Baixa" possui baixo *precision* e alto *recall* ou vice-versa. Para que a comparação seja feita, é aconselhável utilizar a variável F1-Score, e nesse caso em ambos o valor encontrado é o mesmo. Essa variável ajuda a medir o *recall* e a *accuracy* ao mesmo tempo, porque usa a média harmônica e não a média aritmética.

Tabela 6 – Resultados de Treino e Teste (*cross-validation*)

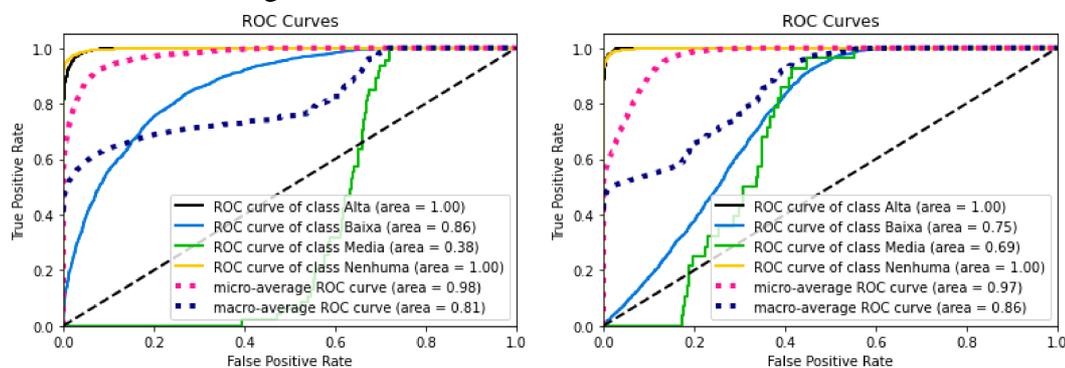
	LBP				Haralick			
	Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
Nenhuma	0.92	0.96	0.94	4545	0.94	0.96	0.95	4570
Baixa	0.67	0.86	0.75	3181	0.65	0.90	0.75	3127
Média	0.11	0.00	0.01	1428	0.32	0.01	0.01	1388
Alta	0.79	0.92	0.85	1801	0.87	0.89	0.88	1870
Accuracy			0.80	10955			0.81	10955
Macro	0.62	0.69	0.64	10955	0.69	0.69	0.65	10955
avg								
Weighted	0.72	0.80	0.75	10955	0.77	0.81	0.76	10955
avg								

Com o objetivo de construir um modelo de predição de risco de câncer de mama para a população feminina chinesa, (GANDOMKAR et al., 2020) coletou 16 variáveis demográficas de um total 1079 mulheres (85 imagens contralaterais a um câncer e 994 casos sem câncer de mama) recrutadas pela *Fudan University Shanghai Cancer Centre*. Um pré-processamento foi realizado e a área densa no tecido foi segmentada de forma automática. Um conjunto de características quantitativas foi extraído da área densa. Ainda no pré-processamento, usando o 80º percentil dos valores de intensidade na área densa, a área segmentada foi limiarizada novamente e o segundo conjunto de características extraídas foi calculado. Os recursos extraídos: demográficos e textura foram usados como entrada em um algoritmo de *Random Forrest 250* e os resultados foram combinados usando *RUSBoost*. (GANDOMKAR et al., 2020) conclui pela AUC de 0,88 do classificador para identificação de imagens de alto risco que o modelo apresentado pode levar ao aumento do poder discriminatório dos modelos de

predição de risco usados atualmente.

Por fim, a curva ROC (AUC) apresentada na Figura 25 mede toda a área bidimensional abaixo da curva é uma medida de quão bem um parâmetro pode atingir entre dois grupos. Frequentemente utilizada em medição para resultados de classificação avalia a 'Taxa de Verdadeiro Positivo' (Sensibilidade / Recuperação) em relação à 'Taxa de Falso Positivo' (1-Especificidade) em diferentes limites. E nas duas apresentadas fica, mais uma vez, evidente o alto índice de falsos positivos encontrados para esta classe.

Figura 25 – Roc Curve de LBP versus Haralick



Fonte: Elaborado pelo autor.

Ma et al. (2019) coletou dados de 331 mulheres chinesas que foram diagnosticadas com câncer de mama invasivo em 2015 e usou 39 características radiômicas quantitativas e morfológicas, *graph features* e *texture features* foram extraídas de área segmentada da lesão. Três classificações binárias foram realizadas: triplo negativo vs não triplo negativo, HER2 vs não HER2 e luminal vs não luminal. O algoritmo de classificação utilizado foi *Naive Bayes* e o desempenho do modelo foi avaliado pela ROC Curve e acurácia. Ma et al. (2019) apresenta mais de uma abordagem mas a que apresentou melhor desempenho geral foi a que utilizou a combinação das extrações de características e não uma das duas somente. O modelo apresentou os seguintes desempenhos de ROC Curve e acurácia de 0,865 e 0,796 para triplo negativo vs não triplo negativo, 0,784 e 0,748 para HER2 vs não HER2, e 0,752 e 0,788 para luminal vs não luminal.

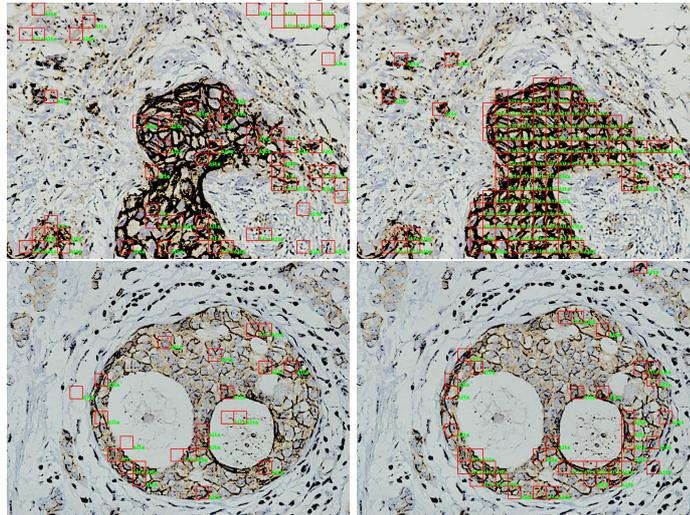
Singh e Kumar (2020) utiliza os mesmos algoritmos de extração de características que este trabalho. O seu trabalho, segue a mesma abordagem: imagem bruta, pré-processamento, extração de características, treinamento, teste e classificação. É utilizado os seguintes tipos de extração de característica: *texture features*, *graph features* e *morphological features*. E dentre os utilizados estão LBP e *Haralick*. Interessante é que no processo de pré-processamento, (Singh; Kumar, 2020) além da transformação de RGB para *Gray Scale* é realizado uma ampliação das imagens. Cada ampliação no conjunto de dados original corresponde a um novo

conjunto de dados, ou seja, foram feitas quatro ampliações então foram gerados quatro novos conjuntos de dados, chamados: 40X, 100X, 200X e 400X. O desempenho do modelo também é avaliado por meio da *accuracy*, *precision*, *recall* e *F1-score*. Todos os vetores de recursos de imagem e morfológicos são utilizados em mais de um classificador e o que apresentou melhor desempenho foi um *cubic SVM* com 92,3% de acurácia.

Löfstedt (2019) em seu trabalho promove uma mudança no algoritmo de textura *Haralick*. Para ser calculado, os recursos de textura *Haralick* dependem do processo de quantização, isto é, precisam que os níveis de cinza da imagem sejam reduzidos. Portanto, sem a realização da quantização os recursos de textura *Haralick* não são reproduzíveis. A mudança proposta pelo autor é desenvolver características de textura *Haralick* que sejam invariantes ao número do nível de cinza da quantização.

A seguir, é apresentado o modelo validando, ou seja, detectando e classificando a hiperexpressão da membrana celular em uma das 12 imagens de 1360 x 1024 pixels que não fazem parte do conjunto de dados de treino e teste. A Figura 24 mostra o resultado da classificação da hiperexpressão a partir de uma SVM utilizando os algoritmos, apresentados neste trabalho, de extração de vetor de textura. Para as 12 imagens de validação, o uso de *Haralick* foi superior ao LBP como mostram todas as métricas apresentadas.

Figura 26 – Validação da Imagem Inteira de LBP *versus* Haralick

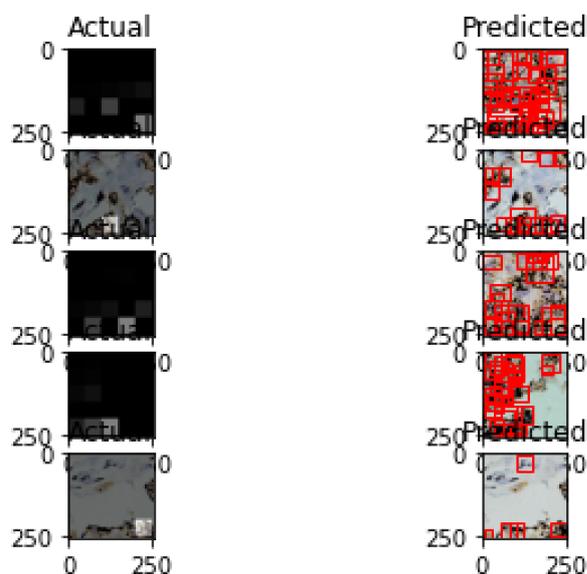


Fonte: Elaborado pelo autor.

Após treino das 25 épocas do conjunto de dados de IHQ pela arquitetura Mask R-CNN, a época que mostrou melhor desempenho, ou seja, uma perda contínua para as *bounding boxes* sobre cada época foi a do arquivo "*maskrcnnimmunohqcfg0013.h5*" que representa a época de número 13. Portanto, este arquivo foi o utilizado para fazer a avaliação do modelo. A avaliação de desempenho do modelo é medida por mAP, mAR, *precisions*, *recalls* e *F1-score*.

A biblioteca Mask R-CNN fornece pacotes e métodos, como *utils* e *computeap* para calcular o AP (*Average Precision*) e outras métricas para uma determinada imagem. Essas pontuações de AP podem ser coletadas em um conjunto de dados e a média calculada para dar uma ideia de quão bom é o modelo na detecção de objetos em um conjunto de dados. Primeiro, foi definido um novo objeto *Config*, como mostra no capítulo Anexo, que foi utilizado para fazer inferências, em vez de treinamento como mostra a Figura 25. Foi estendido a classe *ImunoHQConfig* definido anteriormente para reutilizar os parâmetros. Em vez disso, foi definido um novo objeto com os mesmos valores para manter o código compacto. A configuração alterou alguns dos padrões de uso da GPU para inferência que são diferentes de como são configurados para treinar o modelo.

Figura 27 – Imagem atual vs Imagem com predição *bounding box*



Fonte: Elaborado pelo autor.

O Mask R-CNN utilizando da técnica de TL para o conjunto de dados de IHQ não resultou em uma pontuação do mAP boa em ambos os conjuntos de dados, treino e teste. O valor encontrado para ambos foi de 0,191 e 0,163 respectivamente. Os valores encontrados para mAR de treino e teste foram 0,072 e 0,112; e valores de *f1-score* de treino e teste foram 0,104 e 0,132. Abaixo a Figura 27 mostra que apesar do desempenho ruim do modelo para o conjunto de dados de IHQ nas etapas de treino e teste, algumas hiperexpressões foram detectadas utilizando $\text{IoU} \geq 70\%$. A imagem possui as áreas destacadas juntamente com seus respectivos índices de confiança encontrados.

Os hiperparâmetros foram ajustados corretamente mas alguns fatores podem ter contribuído para um desempenho ruim do uso da arquitetura Mask R-CNN ao cenário deste trabalho: um conjunto de dados extremamente pequeno para redes profundas e totalmente

ICPR 2012 e 0,84 de *precision*, 0,58 de *recall* e 0,69 de *f1-score* para o conjunto de dados ICPR 2014. A capacidade de generalização da abordagem proposta foi testada em outro conjunto de dados, o TUPAC16 (*tumor proliferation assessment challenge 2016*) e o resultado foi de bom desempenho, atestando a capacidade de generalização da abordagem. A abordagem consiste em um treinamento em dois estágios diferentes: o primeiro o treino é feito pela Faster R-CNN; e o segundo pela ResNet-50 e DenseNet-201.

Ainda em (MAHMOOD et al., 2020), as amostras de *true positive* e *false positive* após passarem por um HOG no primeiro estágio de treino, servem como entrada para a ResNet-50 e DenseNet-201 respectivamente, iniciando o segundo estágio da abordagem proposta. E a saída, a classificação das células mitóticas é realizada por meio de um *score-level fusion* da Resnet-50 e DenseNet-201.

Qual modelo tem a maior precisão, objetividade e reprodutibilidade? Não é tão fácil ver se um algoritmo é melhor do que outro apenas analisando os valores de taxa de erro e precisão, uma vez que não existe um algoritmo de classificação para todos os desafios a serem superados. É importante entender o poder e as limitações dos diferentes classificadores e algoritmos de extração de recursos, e que é um desafio poder usá-los da melhor maneira possível, a fim de comparar os modelos em questão. Existem boas análises sobre a comparação de algoritmos de classificação. Talvez o mais importante, o trabalho comparativo deve ser feito em uma estrutura estatisticamente aceitável (SALZBERG, 1997). O desafio de comparar algoritmos em IA visa encontrar uma abordagem de propósito geral que possa raciocinar com modelos estocásticos expressivos (DIETTERICH, 1997).

6 CONCLUSÃO

O primeira abordagem proposta é projetada para classificação binária de imagens de exames de imuno-histoquímica de câncer de mama. O modelo envolve pré-processamento, extração de recursos, treinamento e teste e classificação do conjunto de dados privado fornecido por patologistas da UFCSPA e possui o TCLE de todos os pacientes envolvidos. O primeiro modelo apresentado envolve uma SVM, portanto, uma maior ênfase é dada à extração de características que desempenham um papel vital na precisão da classificação. Para a extração de características foi utilizado dois algoritmos: LBP e *Haralick*. Alguns pesquisadores já abordaram a extração de recursos juntamente com modelos de aprendizado de máquina em conjunto de dados histopatológicos mas na abordagem proposta ilustramos com o conjunto de dados de exames de imuno-histoquímica privado. O desempenho do modelo proposto é avaliado por meio de *accuracy*, *precision*, *recall*, *f1-score* e *Roc Curve*. A acurácia foi de 81% com um classificador Linear SVC utilizando o algoritmo de extração de característica *Haralick*, e 80% de acurácia utilizando o algoritmo de extração de característica LBP. O segunda abordagem consistiu no uso do *framework* Mask R-CNN para o mesmo conjunto de dados da primeira abordagem, porém, reduzido em 162 imagens de IHQ. O modelo utilizou da configuração de alguns hiperparâmetros específicos do *framework* utilizado e também um número expressivo de épocas para treinamento, 25, onde foram 137 imagens para treino e 25 para teste. Trabalhos recentes já usaram o mesmo *framework*, porém, o foco da classificação foi de objeto e não de textura como é o caso deste trabalho. Por fim, o desempenho do modelo que usou como *backbone* uma ResNet-50 também foi avaliado por meio de *accuracy*, *precision*, *recall* e *f1-score*. A mAP de treino foi de 0,19 e o mAP de teste foi de 0,16.

Todo o trabalho foi feito em colaboração com os médicos da UFCSPA e utilizou como norte, o trabalho de (FERNANDES; ROHE, 2016). Entretanto, o mesmo não preservou os rótulos utilizados para extração de valores que resultou nas métricas obtidas. Seria de muita valia se o mesmo tivesse ocorrido, pois, teríamos um modelo para comparação já que este trabalho não esteve presente in situ ao laboratório de patologia. Todo o processo de detecção e criação dos rótulos nas imagens foi feito através de um algoritmo que utiliza um range de valores que detecta e rotula as 4 classes. Após, os patologistas apresentaram feedback quanto sua acurácia em alguns patches. Seria interessante que essa etapa fosse feita exclusivamente pelos patologistas. O que poderia trazer uma melhor performance para a classe "Média". O trabalho possui foco na textura e como o modelo de rede neural profundo utilizado possui característica de detecção de objetos, pode não ter apresentado um desempenho esperado para a detecção da hiperexpressão da membrana celular, pois, a mesma não possui forma definida.

Como trabalho futuro ainda no foco da hiperexpressão da membrana celular a partir da

textura em imagens de IHQ é interessante usar os dois algoritmos de extração de características, *LBP* e *Haralick*, utilizados neste trabalho juntamente com os dados clínicos em um classificador como SVM. Ainda neste trabalho, devido a limitação da detecção e classificação da hiperexpressão baseado na textura de imagens IHQ com o uso de uma RNC como a Mask R-CNN, uma outra possibilidade futura é de usar o mesmo conjunto de dados para detecção e localização da e-cad nuclear que consiste em detectar núcleos celulares hiperexpressos. Desta forma, o trabalho não teria foco na textura para detecção mas inicialmente na forma e, posteriormente, na textura encontrada no núcleo celular.

6.1 Contribuições

O modelo tenta ser capaz de preencher as lacunas propondo uma queda na subjetividade de quantificação e uma maior agilidade no processo de diagnóstico clínico quando do envolvimento da técnica de imuno-histoquímica em relação ao estudo da proteína e-caderina. Ademais, o modelo proposto reproduz o trabalho diário de um profissional de patologia, isto é, o cerne de todo o trabalho é o estudo celular de forma objetiva, sem viés que possa auxiliar o estadiamento de pacientes.

REFERÊNCIAS

- ABDULLA, Waleed. **Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow**. [S.l.]: Github, 2017. <https://github.com/matterport/Mask_RCNN>.
- AGRAVAT, Rupal R. et al. In: DEY, Nilanjan et al. (Ed.). **Soft Computing Based Medical Image Analysis**. Academic Press, 2018. Disponível em: <<http://www.sciencedirect.com/science/article/pii/B9780128130872099875>>.
- AKKUS, Zeynettin et al. **Deep Learning for Brain MRI Segmentation: State of the Art and Future Directions**. [S.l.]: Springer New York LLC, 2017. 449–459 p.
- AL-MASNI, Mohammed A et al. **Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system**. 2018. 85–94 p. Disponível em: <www.elsevier.com/locate/cmpb>.
- ALESKANDARANY, Mohammed A. et al. **Epithelial mesenchymal transition in early invasive breast cancer: An immunohistochemical and reverse phase protein array study**. [S.l.]: Springer New York LLC, 2014. 339–348 p.
- ANDERS, Carey K et al. **Young Age at Diagnosis Correlates With Worse Prognosis and Defines a Subset of Breast Cancers With Shared Patterns of Gene Expression**. 2008. Disponível em: <www.biostat.harvard.edu/wjohnson/ComBat/ComBat.html>.
- ASRI, Hiba et al. **Using machine learning algorithms for breast cancer risk prediction and diagnosis**. [S.l.]: Elsevier, 2016. 1064–1069 p.
- ASWATHY, MA; JAGANNATH, M. **Detection of breast cancer on digital histopathology images: Present status and future possibilities**. 2017. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S235291481630034X>>.
- BEJNORDI, Babak Ehteshami et al. **Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer**. [S.l.]: American Medical Association, 2017. 2199–2210 p.
- CANCELLO, G et al. **Prognosis and adjuvant treatment effects in selected breast cancer subtypes of very young women (< 35 years) with operable breast cancer**. 2010. Disponível em: <<https://academic.oup.com/annonc/article-abstract/21/10/1974/317169>>.
- CHANG, Chih-Chung; LIN, Chih-Jen. **LIBSVM: a library for support vector machines**. [S.l.]: Acm, 2011. 27 p.
- CHEN, Dechang et al. **An algorithm for expanding the TNM staging system**. [S.l.]: Future Medicine Ltd., 2016. 1015–1024 p.
- CHOI, Yoomi et al. **Epithelial-mesenchymal transition increases during the progression of in situ to invasive basal-like breast cancer**. 2013. 2581–2589 p.
- COELHO, Luis Pedro. **Mahotas: Open source software for scriptable computer vision**. 2013.
- De Sousa E Melo, Felipe et al. **Poor-prognosis colon cancer is defined by a molecularly distinct subtype and develops from serrated precursor lesions**. [S.l.]: Nature Medicine, 2013. 614–618 p.

DEMIRKAN, Binnaz. **Clinical Medicine The Roles of Epithelial-to-Mesenchymal Transition (EMT) and Mesenchymal-to-Epithelial Transition (MET) in Breast Cancer Bone Metastasis: Potential Targets for Prevention and Treatment.** OPEN ACCESS J. Clin. Med, 2013. 264–282 p. Disponível em: <www.mdpi.com/journal/jcm>.

DENKER, John S; LECUN, Yann. **Transforming Neural-Net Output Levels to Probability Distributions.** [S.l.], 1991.

DIETTERICH, Thomas G. **Machine-Learning Research.** AI Magazine, 1997. 97–136 p. Disponível em: <<http://www.aaai.org/ojs/index.php/aimagazine/article/view/1324>>.

DONG, Fei et al. **Computational Pathology to Discriminate Benign from Malignant Intraductal Proliferations of the Breast.** PLoS ONE, 2014. e114885 p. Disponível em: <<https://dx.plos.org/10.1371/journal.pone.0114885>>.

DUA, D et al. **UCI Machine Learning Repository. Irvine, CA: University of California.** 2017. Disponível em: <<http://archive.ics.uci.edu/ml>>.

ESTEVA, Andre et al. **Dermatologist-level classification of skin cancer with deep neural networks.** [S.l.]: Nature, 2017. 115–118 p.

FERLAY, Jacques et al. **Cancer incidence and mortality worldwide: Sources, methods and major patterns in GLOBOCAN 2012.** [S.l.]: International Journal of Cancer, 2015. E359–E386 p.

FERNANDES, Ana Paula da Silva; ROHE, Adriana. **Estudo da expressão imuno-histoquímica da e-caderina e n-caderina em câncer da mama.** 2016.

Fred Agarap, Abien M. **Deep Learning using Rectified Linear Units (ReLU).** [S.l.], 2019. Disponível em: <<https://github.com/AFAgarap/relu-classifier>>

GANDOMKAR, Ziba et al. **Breast cancer risk prediction in Chinese women based on mammographic texture and a comprehensive set of epidemiologic factors.** SPIE, 2020. 257 – 262 p. Disponível em: <<https://doi.org/10.1117/12.2564276>>.

GARCIA, Laura M et al. **A study of the suitability of autoencoders for preprocessing data in breast cancer experimentation.** [S.l.]: Journal of Biomedical Informatics, 2017.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning.** [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>.

GRAHAM, Benjamin. **Fractional Max-Pooling.** 2014. Disponível em: <<http://arxiv.org/abs/1412.6071>>.

GUNASINGHE, N. P.A.Devika et al. **Mesenchymal-epithelial transition (MET) as a mechanism for metastatic colonisation in breast cancer.** [S.l.]: Cancer and Metastasis Reviews, 2012. 469–478 p.

HAMMOND, M Elizabeth H et al. **College of American Pathologists Conference XXXV : Solid Tumor Prognostic Factors — Which , How and So What ?** [S.l.]: Archives of pathology & laboratory medicine, 1999. 958–65 p.

HAN, Jiawei; PEI, Jian; KAMBER, Micheline. **Data mining: concepts and techniques.** [S.l.]: Elsevier, 2011.

HAYKIN, S.; HAYKIN, S.S. **Neural Networks and Learning Machines**. Prentice Hall, 2009. (Neural networks and learning machines, v. 10). Disponível em: <https://books.google.com/books?id=K7P36IKzI_QC>.

HE, Kaiming et al. **Mask R-CNN: Proceedings of the IEEE International Conference on Computer Vision (ICCV)**. 2017.

Heikkila, M.; Pietikainen, M. **A texture-based method for modeling the background and detecting moving objects**. [S.l.]: IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006. 657-662 p.

HOMAN, SG et al. **Peer reviewed: risk factors, preventive practices, and health care among breast cancer survivors, United States, 2010**. ncbi.nlm.nih.gov, 2016. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4722934/>>.

HSU, Chih-Wei; LIN, Chih-Jen. **A Comparison of Methods for Multiclass Support Vector Machines**. [S.l.], 2002. v. 13, n. 2, 415 p.

INCA. **ABC do câncer: abordagens básicas para o controle do câncer / Instituto Nacional de Câncer**. [S.l.]: Inca, 2011.

IRSHAD, Humayun et al. **Methods for nuclei detection, segmentation, and classification in digital histopathology: A review-current status and future potential**. [S.l.]: Institute of Electrical and Electronics Engineers, 2014. 97–114 p.

JAIN, Aashna et al. **Cancerous Cell Detection Using Histopathological Image Analysis**. International Journal of Innovative Research in Computer and Communication Engineering (An ISO, 2014. Disponível em: <www.ijirce.com>.

JONES, Jacqueline et al. **Nuclear localization of Kaiso promotes the poorly differentiated phenotype and EMT in infiltrating ductal carcinomas**. [S.l.]: Clinical and Experimental Metastasis, 2014. 497–510 p.

KARPATHY, A. **Convolutional Neural Networks for Visual Recognition**. 2017. Disponível em: <<http://cs231n.github.io/convolutional-networks/{#}fc>>.

KHOSRAVI, Pegah et al. **Deep Convolutional Neural Networks Enable Discrimination of Heterogeneous Digital Pathology Images**. [S.l.]: Elsevier B.V., 2018. 317–328 p.

KIM, I et al. **A predictive model for high/low risk group according to oncotype DX recurrence score using machine learning**. Elsevier, 2019. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0748798318314306>>.

KNERR, S.; PERSONNAZ, L.; DREYFUS, G. **Single-layer learning revisited: a stepwise procedure for building and training a neural network**. [S.l.]: Springer Berlin Heidelberg, 1990. 41–50 p.

KYLE, Robert A.; RAJKUMAR, S. Vincent. **Epidemiology of the plasma-cell disorders**. 2007. 637–664 p.

LECUN, Y; BENGIO, Y; HINTON, G. **Deep Learning & Convolutional Networks**. [S.l.]: IEEE, 2015.

LONG, Jonathan; SHELHAMER, Evan; DARRELL, Trevor. **Fully Convolutional Networks for Semantic Segmentation**. [S.l.]: IEEE, 2015.

LONG, Mingsheng et al. **Learning Transferable Features with Deep Adaptation Networks**. [S.l.], 2015.

LÖFSTEDT, Tommy et al. **Gray-level invariant Haralick texture features**. Public Library of Science, 2019. 1–18 p. Disponível em: <<https://doi.org/10.1371/journal.pone.0212110>>.

MA, Li; WEINBERG, Robert A. **Micromanagers of malignancy: role of microRNAs in regulating metastasis**. [S.l.]: Elsevier, 2008.

MA, Wenjuan et al. **Breast Cancer Molecular Subtype Prediction by Mammographic Radiomic Features**. 2019. 196 - 201 p. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1076633218300527>>.

MACÍAS-GARCÍA, Laura et al. **A study of the suitability of autoencoders for preprocessing data in breast cancer experimentation**. 2017. 33 - 44 p. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1532046417301478>>.

MAHMOOD, Tahir et al. **Artificial Intelligence-Based Mitosis Detection in Breast Cancer Histopathology Images Using Faster R-CNN and Deep CNNs**. Journal of Clinical Medicine, 2020. Disponível em: <<https://www.mdpi.com/2077-0383/9/3/749>>.

Martins Da Rosa, Luciana; RADÜNZ, Vera. **Taxa de sobrevida na mulher com câncer de mama: Estudo de revisão**. 2012.

MASUDA, T et al. **DNA binding region of BRCA1 affects genetic stability through modulating the intra-S-phase checkpoint**. ncbi.nlm.nih.gov, 2016. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4737671/>>.

MITCHELL, T.M. **Machine Learning**. McGraw-Hill, 1997. (McGraw-Hill International Editions). Disponível em: <<https://books.google.com/books?id=EoYBngEACAAJ>>.

MONTAZERI, Mitra et al. **Machine learning models in breast cancer survival prediction**. [S.l.]: IOS Press, 2016. 31–42 p.

MORENO-BAREA, FJ et al. **Forward Noise Adjustment Scheme for Data Augmentation**. IEEE, 2018. Disponível em: <<https://ieeexplore.ieee.org/abstract/document/8628917/>>.

MOSQUERA-LOPEZ, Clara; ESCOBAR, Rodrigo; AGAIAN, Sos. **Modeling human-perceived quality for the assessment of digitized histopathology color standardization**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2015.

Nacional De Câncer, Instituto; Gomes Da Silva, José Alencar. **Ministério da Saúde Diretrizes para a Detecção Precoce do Câncer de Mama no Brasil**. [S.l.], 2015.

Ojala, T.; Pietikainen, M.; Maenpaa, T. **Multiresolution gray-scale and rotation invariant texture classification with local binary patterns**. [S.l.]: IEEE, 2002. 971-987 p.

PAN, Sinno Jialin; YANG, Qiang. **A survey on transfer learning**. 2010. 1345–1359 p.

PARIKH, R et al. **Understanding and using sensitivity, specificity and predictive values**. 2008. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2636062/>>.

- PONTI, Moacir; COSTA, Gabriel da. **Como funciona o Deep Learning**. [S.l.: s.n.], 2018.
- REZENDE, S.O. **Sistemas inteligentes: fundamentos e aplicações**. Manole, 2003. Disponível em: <https://books.google.com/books?id=UsJe_PlbnWcC>.
- ROFFO, Giorgio. **Ranking to Learn and Learning to Rank: On the Role of Ranking in Pattern Recognition Applications**. 2017.
- SAINATH, Tara N. et al. **Deep convolutional neural networks for LVCSR**. [S.l.]: IEEE, 2013. 8614–8618 p.
- SALZBERG, Steven. **On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach**. 1997. 317–328 p. Disponível em: <<https://doi.org/10.1023/A:1009752403260>>.
- SHORTEN, Connor; KHOSHGOFTAAR, Taghi M. **A survey on Image Data Augmentation for Deep Learning**. [S.l.]: SpringerOpen, 2019.
- SIEGEL, Rebecca L.; MILLER, Kimberly D.; JEMAL, Ahmedin. **Cancer statistics**. [S.l.]: Wiley, 2017. 7–30 p.
- Singh, S.; Kumar, R. **Histopathological Image Analysis for Breast Cancer Detection Using Cubic SVM**. 2020. 498-503 p.
- SOCIETY, American Cancer. **Breast Cancer Facts Figures 2013-2014**. 2013. Disponível em: <<https://www.cancer.org/research/cancer-facts-statistics/breast-cancer-facts-figures.html>>.
- STEINESTEL, Konrad et al. **Clinical significance of epithelial-mesenchymal transition**. [S.l.]: Springer Nature, 2014. 17 p.
- STEWART, WILD. **World Cancer Report**. 2014. Disponível em: <<https://shop.iarc.fr/products/world-cancer-report-2014>>.
- SUN, J et al. **Associations and indications of Ki67 expression with clinicopathological parameters and molecular subtypes in invasive breast cancer: A population-based**. Disponível em: <<https://www.spandidos-publications.com/ol/10/3/1741>>.
- SYLVA, Eliane da. **Ministério da Saúde Instituto Nacional de Câncer José Alencar Gomes da Silva (INCA)**. 2017.
- VAPNIK, Vladimir. **Statistical learning theory**. 1998. [S.l.]: Wiley, New York, 1998.
- WILLETT, WC et al. **Epidemiology and assessing and managing risk**. 2000.
- YU, Kun Hsing et al. **Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features**. [S.l.]: Nature Publishing Group, 2016.
- ZHANG, Yu Dong; MUHAMMAD, Khan; TANG, Chaosheng. **Twelve-layer deep convolutional neural network with stochastic pooling for tea category classification on GPU platform**. [S.l.]: Springer New York LLC, 2018. 22821–22839 p.
- ZHONG, Zhun et al. **Random Erasing Data Augmentation**. 2017. Disponível em: <<http://arxiv.org/abs/1708.04896>>.
- ZHU, Wentao et al. **Deep multi-instance networks with sparse label assignment for whole mammogram classification**. [S.l.]: Springer Verlag, 2017. 603–611 p.

7 ANEXOS

Código fonte em Python do Modelo 1

Python

Implementação do Modelo 1

```

import sys import resource from sklearn.svm import LinearSVC, SVC from
sklearn.model_selection import train_test_split, learning_curve, cross_val_score, KFold import cv2 import random
accuracy from sklearn.metrics import accuracy_score from sklearn.metrics import precision, recall
class Supervised:
    def
    init(self, name, trainFolder, validationFolder, extImage, dy, dx): self.name = name self.trainFolder = trainFolder self.validationFolder = validationFolder
    def normalizaImage(self, image): image = image.astype("float") image /= (image.sum() +
1e-7) return image
    def std(self, features, shift=100000): lstStd = [] tot = len(features) - 1 print(tot) for i in
range(0, tot, shift): if (tot - i) < shift: break else: print("Std", np.std(features[i:i+shift]))
lstStd.append(np.std(features[i:i+shift])) return lstStd
    def pretreino(self, metodo=3): Inicializando a matriz de dados e a lista de labels data = []
labels = [] cont=0 Loop pelas imagens de treino que são 50x50 pixels e rotuladas for imageTrain
in glob.glob(self.trainFolder + "/*" + self.extImage):
    cont += 1 if cont == 10: break
    image = cv2.imread(imageTrain) image = cv2.cvtColor(image,
cv2.COLOR_BGR2GRAY) texture = imageTrain.split(",")[3]
    print("Gerando vetor de recursos da imagem", cont, imageTrain, "... ")
    if metodo == 1: Extrai as características de textura de Haralick em 4 direções, depois
calcula a média de cada direção features = mahotas.features.haralick(image,
distance=3).mean(axis=0) elif metodo == 2: features =
self.std(mahotas.features.lbp(self.normalizaImage(image), self.r,
self.n_points, ignore_zeros = True)) features =
mahotas.features.lbp(self.normalizaImage(image), self.r, self.n_points, ignore_zeros =
True) else : lstMean, lstStd = lbp.main(image) features = lstStd
    atualiza data e label data.append(features) labels.append(texture)
    return data, labels
    def validacao(self, model, algorithm, metodo=3): print("Validando o modelo...") cont = 0
Loop pelas imagens de validação que são 1036x1024 pixels e não possuem rotulos for
imageTest in glob.glob(self.validationFolder + "/*" + self.extImage): print(imageTest) img =
cv2.imread(imageTest) image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

```

For para percorrer a imagem em pedaços de 50x50 pixels for y in range(0, image.shape[0], self.dy): for x in range(0, image.shape[1], self.dx): Garantindo que o pedaço tem 50x50 pixels pedaco = image[y: y+self.dy, x: x+self.dx] if(pedaco.shape[0] == self.dy and pedaco.shape[1] == self.dx):

```

    if metodo == 1: features = mahotas.features.haralick(pedaco, distance=3).mean(axis=0)
    elif metodo == 2: features = svm.localBinaryPatterns(pedaco, 24, 8) features =
self.std(mahotas.features.lbp(self.normalizaImage(pedaco), 3, 24).std(axis=0)) features =
mahotas.features.lbp(self.normalizaImage(pedaco), 3, 24).std(axis=0) else: lstMean, lstStd =
lbp.main(pedaco) features = lstStd

```

```

    print(features.reshape(1, -1), , features) pred = model.predict(features.reshape(1, -1))[0]
print("Predição", pred)

```

```

    if pred == 'Alta': cv2.rectangle(img, (x+self.dx, y), (x, y+self.dy), (0, 0, 255, 255), 2)
cv2.putText(img, str(pred), ((x+self.dx), (y+self.dy)),
cv2.FONT_HERSHEY_COMPLEX_SMALL, 1.0, (0, 255, 0), 2)

```

```

    cv2.imwrite(self.validationFolder + 'Result'+algorithm+str(cont)+' .png',img)cont+=1

```

```

def localBinaryPatterns(self, image, numPoints, radius, eps=1e-7, method='uniform'): lbp
= feature.local_binary_pattern(image, numPoints, radius, method)

```

```

    (hist, ) = np.histogram(lbp.ravel(), bins = range(0, numPoints + 3), range =
(0, numPoints + 2))

```

```

    hist = hist.astype("float") hist /= (hist.sum() + eps)

```

```

    return hist

```

```

def setAccuracyScore(self, y_test, y_pred, title) :
print("
=====
", title, "
=====
")vl_score = accuracy_score(y_test, y_pred)print(" Accuracy :
self.plotConfusionMatrix(y_test, y_pred, False)self.plotConfusionMatrix(y_test, y_pred, True)return

```

```

def setAUCScore(self, clsf, y_test, X_test) : fpr, tpr, thresholds =
roc_curve(y_test, clsf.decision_function(X_test), pos_label = 'Alta')print(" Val fpr ==>
", fpr, " Val tpr ==> ", tpr, " Val Thresholds ==> ", thresholds)print(" fpr ==>
", fpr.mean(), " tpr ==> ", tpr.mean(), " Thresholds ==> ", thresholds.mean())vl_auc =
auc(fpr, tpr)print(" AUC :

```

```

class_names = ["Alta", "Nenhuma"]fpr = dict()tpr = dict()thr = dict()roc_auc =
dict()

```

```

for i in range(len(class_names)) : fpr[i], tpr[i], thr[i] = roc_curve(y_test[:
, i], clsf.decision_function(X_test)[: , i])roc_auc[i] =
auc(fpr[i], tpr[i])print(" fpr", fpr[i], " tpr", tpr[i], " thresholds", thr[i])self.plotRocCurve(roc_auc, fpr

```

```

def plotRocCurve(self, rocauc, fpr, tpr): plt.figure() lw=2

```

```

plt.plot(fpr[2], tpr[2], color='darkorange', lw=lw, label='ROC Curve (area = plt.plot([0, 1],
[0, 1], color='navy', lw=lw, linestyle='--') plt.xlim([0.0, 1.0]) plt.ylim([0.0, 1.05])

```

```
plt.xlabel("False Positive Rate") plt.ylabel("True Positive Rate") plt.title("Receiver Operating
Characteristic - ROC") plt.legend(loc="lower right") plt.show()
```

```
def plotConfusionMatrix(self, y_test, y_pred, normalize = True) : class_names =
["Alta", "Nenhuma"] cm = confusion_matrix(y_test, y_pred) cmap =
plt.cm.Bluesprint(" Matriz de Confusão", cm) print(" Report =====>
", classification_report(y_test, y_pred), "", precision_recall_fscore_support(y_test, y_pred))
```

```
if normalize: cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis] print("Matriz de
confusão normalizada") else: print("Matriz de confusão sem normalização")
```

```
print("", cm)
```

```
plt.imshow(cm, interpolation='nearest', cmap=cmap) plt.title("Matriz de Confusão")
plt.colorbar()
```

```
tick_marks = np.arange(len(class_names)) plt.xticks(tick_marks, class_names, rotation =
45) plt.yticks(tick_marks, class_names)
```

```
fmt = '.2f' if normalize else 'd' thresh = cm.max() / 2. for i, j in
itertools.product(range(cm.shape[0]), range(cm.shape[1])): plt.text(j, i, format(cm[i, j], fmt),
horizontalalignment="center", color="white" if cm[i, j] > thresh else "black")
```

```
plt.ylabel("True Label") plt.xlabel("Predicted Label")
plt.tight_layout() plt.figure() plt.show()
```

```
Plot non-normalized confusion matrix plt.figure() setConfusionMatrix(cm,
classes=class_names, title = '
Confusion matrix, without normalization') Plot normalized confusion matrix plt.figure() setConfusionMatrix(cm,
class_names, normalize = True, title = ' Normalized confusion matrix') plt.show()
```

```
def plotLearningCurve(self, estimator, title, X, y, ylim=None, cv=None, n_jobs =
1, train_sizes = np.linspace(.1, 1.0, 5)) : plt.figure() plt.title(title) if ylim is not None :
plt.ylim(*ylim) plt.xlabel("Training Dataset") plt.ylabel("Score")
```

```
train_sizes, train_scores, test_scores = learning_curve(estimator, X, y, cv = cv, n_jobs =
n_jobs, train_sizes = train_sizes)
```

```
train_scores_mean = np.mean(train_scores, axis = 1) train_scores_std =
np.std(train_scores, axis = 1) test_scores_mean = np.mean(test_scores, axis =
1) test_scores_std = np.std(test_scores, axis = 1)
```

```
print("Train Sizes", train_sizes, "Estimator", estimator)
```

```
plt.fill_between(train_sizes, train_scores_mean - train_scores_std, train_scores_mean +
train_scores_std, alpha = 0.1, color = "r") plt.fill_between(train_sizes, test_scores_mean -
test_scores_std, test_scores_mean + test_scores_std, alpha = 0.1, color =
"g") plt.plot(train_sizes, train_scores_mean, 'o-', color = "r", label =
"Training Score") plt.plot(train_sizes, test_scores_mean, 'o-', color = "g", label =
"Cross - Validation Score")
```

```
plt.legend(loc="best") plt.show()
```

```
def plotBoxPlot(self, resultAccuracy, names): fig = plt.figure() fig.subtitle("Accuracy") ax
```

```

=
fig.add_subplot(111)plt.boxplot(resultAccuracy)ax.set_xlabel(names)ax.set_ylabel("AccuracyScore")
def memoryLimit(percentage=0.9):
    soft, hard = resource.getrlimit(resource.RLIMIT_AS)resource.setrlimit(resource.RLIMIT_AS, (getMemory()*
1024 * percentage, hard))
    def getMemory():
        with open('/proc/meminfo', 'r') as mem:
            free_memory = 0
            for i in mem:
                sline = i.split()
                if sline[0] in ('MemFree:', 'Buffers:', 'Cached:'):
                    free_memory += int(sline[1])
            return free_memory
    Argumentos = ap = argparse.ArgumentParser()
    ap.add_argument("-d", "--training", required = True, help =
"Caminho para o diretorio do dataset de treino")
    ap.add_argument("-v", "--validation", required = True, help =
"Caminho para o diretorio do dataset de validacao")
    ap.add_argument("-l", "--ext", default = '.tif', help =
"Extensao das imagens do dataset")
    ap.add_argument("-y", "--dy", type = int, default = 50, help =
"Tamanho em pixels do pedaco da imagem a ser extraido")
    ap.add_argument("-x", "--dx", type = int, default = 50, help =
"Tamanho em pixels do pedaco da imagem a ser extraido")
    if __name__ == '__main__':
        memoryLimit()
        try:
            args = vars(ap.parse_args())
            svm = Supervised('SVM', args["training"], args["validation"], args["ext"], args["dy"],
args["dx"])
            dx = svm.dx
            dy = svm.dy
            models = []
            models.append((3, 'LSVM', LinearSVC(C=10.0, dual=False,
random_state = 42, max_iter = 1000, verbose = True), 'ScikitLBP'))
            models.append((1, 'LSVM', LinearSVC(C = 10.0, dual = False,
random_state = 42, max_iter = 1000, verbose = True), 'Haralick'))
            resultAccuracy = []
            names = []
            for number, name, model, algorithm in models:
                print("Carregando Dataset...")
                data, labels = svm.pretreino(number)
                X_train, X_test, y_train, y_test = train_test_split(data, labels, train_size = 0.7, test_size =
0.3)
                Treinando o modelo print("Treinando o modelo...")
                cache_size = 200, max_iter = 1000, verbose = True)
                try:
                    model.fit(X_train, y_train)
                    kfold = KFold(n_splits = 10)
                    accuracy_results = cross_val_score(model, X_test, y_test, cv = kfold, scoring = 'accuracy')
                    resultAccuracy.append(accuracy_results)
                    names.append(name + ' - ' +

```

```

algorithm)accuracyMessage = "print(accuracyMessage)exceptExceptionasexc :
print(" Errooafazer fitdomodelo!", exc)Testandoomodeloprint("Testandoomodelo...")try :
ypred = model.predict(X_test)svm.plotlearningCurve(model," LinearSVCwith" +
algorithm, X_train, y_train)svm.setAccuracyScore(y_test, y_pred," AcurciaTotal")svm.setAUCScore(
print(" Errooafazeraprediodapartedeteste!", exc)
print("Validando e classificando...") svm.validacao(model, algorithm, number)
svm.plotBoxPlot(resultAccuracy, names) except MemoryError as mer: print("Erro de
memória!", mer) sys.stderr.write(' : Memory Exception') sys.exit(1)

```

Código fonte em Python do Modelo 2

Célula 1 do Jupyter Notebook

```

from os import listdir from xml.etree import ElementTree from numpy import zeros from
numpy import asarray from mrcnn.utils import Dataset from matplotlib import pyplot from
mrcnn.visualize import
display_instancesfrommrcnn.utilsimportextract_boxesfrommrcnn.configimportConfigfrommrcnn
class ImunoIHQDataset(Dataset):
def
load_dataset(self, dataset_dir, is_train = True) : self.add_class(" dataset", 1, "imunoHQ")
images_dir = dataset_dir + ' /images/' annotations_dir = dataset_dir + ' /anns/'
for filename in listdir(images_dir) : image_id = filename[: -4]
if image_idin['00090'] : continue
if is_trainandint(image_id) >= 150 : continue
img_path = images_dir + filenameann_path = annotations_dir + image_id + '.xml'
self.add_image('dataset', image_id = image_id, path = img_path, annotation = ann_path)
def extract_boxes(self, filename) : tree = ElementTree.parse(filename)root =
tree.getroot()boxes = list()forboxinroot.findall('./bndbox') : xmin =
int(box.find('xmin').text)ymin = int(box.find('ymin').text)xmax =
int(box.find('xmax').text)ymax = int(box.find('ymax').text)coors =
[xmin, ymin, xmax, ymax]boxes.append(coors)
width = int(root.find('./size/width').text) height = int(root.find('./size/height').text)
return boxes, width, height
def
load_mask(self, image_id) : info = self.image_info[image_id]path = info['annotation']
boxes, w, h = self.extract_boxes(path)masks = zeros([h, w, len(boxes)], dtype = ' uint8')
class_ids = list()foriinrange(len(boxes)) : box = boxes[i]row_s, row_e =
box[1], box[3]col_s, col_e = box[0], box[2]masks[row_s : row_e, col_s : col_e, i] =
1class_ids.append(self.class_names.index('imunoHQ'))returnmasks, asarray(class_ids, dtype = '

```

```

int32')
def
image_reference(self, image_id) : info = self.image_info[image_id]return info['path']
Define a configuration for the model class ImunoIHQConfig(Config): define the name of
the configuration NAME =
"imunoHQ_cfg" number_of_classes(background + imunoHQ) NUM_CLASSES =
1 + 1 number_of_training_steps_per_epoch STEPS_PER_EPOCH =
10 TRAIN_RATIO_SPER_IMAGE = 50 GPU_COUNT = 0 IMAGE_MIN_DIM =
1024 IMAGE_MAX_DIM = 1360 USE_MINI_MASK = True

```

Célula 2 do Jupyter Notebook

```

train set train_set = ImunoIHQDataset()train_set.load_dataset('imunoHQ', is_train =
True)train_set.prepare()print('Train :
test/val set test_set = ImunoIHQDataset()test_set.load_dataset('imunoHQ', is_train =
False)test_set.prepare()print('Test :
load an image image_id = 0image = train_set.load_image(image_id)print(" ImageShape :
", image.shape)
load image mask mask,
class_ids = train_set.load_mask(image_id)print(" MaskShape : ", mask.shape)
plot image pyplot.imshow(image)
plot mask pyplot.imshow(mask[:, :, 0], cmap='gray', alpha=0.5) pyplot.show()

```

Célula 3 do Jupyter Notebook

```

plot first few images for i in range(9): pyplot.subplot(330 + 1 + i) image =
train_set.load_image(i)pyplot.imshow(image)mask, = train_set.load_mask(i)
for j in range(mask.shape[2]): pyplot.imshow(mask[:, :, j], cmap = 'gray', alpha = 0.3)
pyplot.show()

```

Célula 4 do Jupyter Notebook

```

enumerate all images in the dataset for
image_id in train_set.image_ids : info = train_set.image_info[image_id]print(info)
define image id image_id = 100image = train_set.load_image(image_id)mask, class_ids =
train_set.load_mask(image_id)bbox =
extract_bboxes(mask)display_instances(image, bbox, mask, class_ids, train_set.class_names)

```

Célula 5 do Jupyter Notebook

```

Train Model prepare config config = ImunoIHQConfig() config.display()
define the model model = MaskRCNN(mode='training',
model_dir='./', config = config)
load weights (mscoco) and exclude the
outputlayersmodel.load_weights('mask_rcnn_coco.h5', by_name = True, exclude =
["mrcnn_class_logits", "mrcnn_bbox_fc", "mrcnn_bbox", "mrcnn_mask"])train_weights(outputlayerssor'he
config.LEARNING_RATE, epochs = 3, layers = 'heads')

```

Célula 6 do Jupyter Notebook

```

Inference from numpy import
expand_dims from numpy import mean from mrcnn.util import compute_ap, compute_recall from mrcnn.
define the prediction configuration class PredictionConfig(Config): define the name of the
configuration NAME = "imunoHQ_cfg" USE_MINI_MASK =
False number of classes (background + imunoHQ) NUM_CLASSES =
1 + 1 simplify GPU config GPU_COUNT = 1 IMAGES_PER_GPU = 1
calculate the mAP for model on a given dataset def
evaluate_model(dataset, model, cfg) : APs = list() ARs = list() F1_scores =
list() precisions = list() recalls = list() for image_id in dataset.image_ids :
load_image, boundingboxes and masks for the image id image, image_meta, gt_class_id, gt_bbox, gt_mask =
load_image_gt(dataset, cfg, image_id, use_mini_mask =
False) convert pixel values (e.g. center) scaled_image =
mold_image(image, cfg) convert image into one sample sample
expand_dims(scaled_image, 0) make prediction yhat = model.detect(sample, verbose =
1) extract results for first sampler
yhat[0] calculate statistics, including AP AP, precisions, recalls, overlaps =
compute_ap(gt_bbox, gt_class_id, gt_mask, r["rois"], r["class_ids"], r["scores"], r["masks"]) AR, positive_ids
compute_recall(r["rois"], gt_bbox, iou = 0.7)
F1_scores.append((2 * (mean(precisions) * mean(recalls)))/(mean(precisions) +
mean(recalls)))
store APs.append(AP) ARs.append(AR) precisions.append(precision)
recalls.append(recall) calculate the mean AP across all images mAP = mean(APs) mAR =
mean(ARs)
return mAP, mAR, F1_scores, precisions, recalls
load the train dataset
train_set = ImunoIHQDataset() train_set.load_dataset('imunoHQ', is_train =
True) train_set.prepare() print('Train :

```

```

load the test dataset
test_set = ImunoIHQDataset()test_set.load_dataset('imunoHQ', is_train =
False)test_set.prepare()print('Test :
create config cfg = PredictionConfig()
define the model model = MaskRCNN(mode = 'inference', model_dir = './', config = cfg)
load model weights
model.load_weights('mask_rcnn_imunohq_cfg0003.h5', by_name = True)
evaluate model on training dataset train_mAP, train_mAR, F1_score, precisions, recalls =
evaluate_model(train_set, model, cfg)print("TrainmAP(AveragePrecision) :
print("TrainmAR(AverageRecall) : print("Firstwaycalculatef1 - score : ", F1_score)
F1_score_2 = (2 * train_mAP * train_mAR)/(train_mAP +
train_mAR)print("Secondwaycalculatef1 - scorebyaverage : ", F1_score_2)
plot_precision_recall(train_mAP, precisions, recalls)
evaluate model on test dataset test_mAP, test_mAR, F1_score, precisions, recalls =
evaluate_model(test_set, model, cfg)print("TestmAP(AveragePrecision) :
print("TestmAR(AverageRecall) : print("Firstwaycalculatef1 - score : ", F1_score)
F1_score_2 = (2 * test_mAP * test_mAR)/(test_mAP +
test_mAR)print("Secondwaycalculatef1 - scorebyaverage : ", F1_score_2)
plot_precision_recall(test_mAP, precisions, recalls)
plt.show()

```

Célula 7 do Jupyter Notebook

```

import random from mrcnn.model import load_image_gt from mrcnn import visualize
def get_ax(rows = 1, cols = 1, size = 8) : ax = plt.subplots(rows, cols, figsize =
(size * cols, size * rows))return ax
Test on a random image image_id =
41random.choice(test_set.image_ids)original_image, image_metas, gt_class_id, gt_box, gt_mask =
load_image_gt(test_set, cfg, image_id, use_mini_mask = False)results =
model.detect([original_image], verbose = 1)r =
results[0]visualize.display_instances(original_image, r['rois'], r['masks'], r['class_ids'], test_set.class_names)
get_ax()

```

Célula 8 do Jupyter Notebook

```

Predict new images import cv2
def get_ax(rows = 1, cols = 1, size = 20) : ax = plt.subplots(rows, cols, figsize =
(size * cols, size * rows))return ax
def

```

```

predictSingle(image_path) : load_image(image_id)
image = skimage.io.imread(image)
image = cv2.imread(image_path)
run object detection results = model.detect([image], verbose = 1)
Display results ax = get_ax(1) r = results[0] a =
visualize.display_instances(image, r['rois'], r['masks'], r['class_ids'], test_set.class_names, r['scores'], ax,
ax, title = "Predictions") print("detections :
".format(np.array(test_set.class_names)[det_class_ids]))
show coordinates cont = 1 print("Coordinates from bounding box") for i in (r['rois']):
print(cont, i) cont += 1
im =
"/home/unisinos/MaskRCNN/immunoHQ/validation/13976 - E2.tif" predictSingle(im)

```