

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ
ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIA EM SAÚDE

ISABELA DAMBISKI GOMES DE CARVALHO

INVESTIGAÇÃO DE TÉCNICAS PARA MELHORAR A RESOLUÇÃO DE
IMAGENS DE MICROSCOPIA DE CÉLULAS DE SANGUE PARA
CLASSIFICAÇÃO

CURITIBA

2023

ISABELA DAMBISKI GOMES DE CARVALHO

**INVESTIGAÇÃO DE TÉCNICAS PARA MELHORAR A RESOLUÇÃO DE
IMAGENS DE MICROSCOPIA DE CÉLULAS DE SANGUE PARA
CLASSIFICAÇÃO**

Dissertação apresentada ao Programa de Pós-Graduação em Tecnologia de Saúde (PPGTS) Área de concentração: Bioengenharia, da Escola Politécnica da Pontifícia Universidade Católica do Paraná, como requisito parcial à obtenção do título de mestre em Tecnologia em Saúde.

Orientador: Prof. Dr. Guilherme Nunes Nogueira Neto

Coorientador: Prof. Dr. Sergio Ossamu Ioshii

CURITIBA

2023

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central

Luci

C331i 2023	Carvalho, Isabela Dambiski Gomes de Investigação de técnicas para melhorar a resolução de imagens de microscopia de células de sangue para classificação / Isabela Dambiski Gomes de Carvalho ; orientador: Guilherme Nunes Nogueira Neto ; coorientador: Sergio Ossamu Ioshii. – 2023. 203 f. ; il. : 30 cm Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2023 Bibliografia: f. 191-203 1. Tecnologia médica. 2. Microscopia. 3. Aprendizado do computador. 4. Redes neurais. I. Nogueira Neto, Guilherme Nunes. II. Ioshii, Sergio Ossamu. III. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Tecnologia em Saúde. IV. Título. CDD. 20. ed. – 610.28
---------------	---

Eduarda Wielganczuk – CRB 9/1118



Programa de
**PÓS-GRADUAÇÃO
EM TECNOLOGIA EM SAÚDE
PUCPR**

TERMO DE APROVAÇÃO DE DISSERTAÇÃO Nº 318

A Dissertação de Mestrado intitulada **“INVESTIGAÇÃO DE TÉCNICAS PARA MELHORAR A RESOLUÇÃO DE IMAGENS DE MICROSCOPIA DE CÉLULAS DE SANGUE PARA CLASSIFICAÇÃO”** defendida em sessão pública pelo(a) candidato(a) **Isabela Dambiski Gomes de Carvalho**, no dia **03 de março de 2023**, foi julgada para a obtenção do grau de Mestre em Tecnologia em Saúde, e aprovado em sua forma final, pelo Programa de Pós-Graduação em Tecnologia em Saúde.

BANCA EXAMINADORA:

Prof. Dr. Guilherme Nunes Nogueira Neto – (PUCPR)

Prof. Dr. Alceu de Souza Britto Junior – (PUCPR)

Prof. Dr. Dante Luiz Escuissato - (UFPR)

A via original deste documento encontra-se arquivada na Secretaria do Programa, contendo a assinatura da Coordenação após a entrega da versão corrigida do trabalho.

Curitiba, 09 de maio de 2023

Prof. Dr. Percy Nohama
**Coordenador do Programa de
Pós-Graduação em Tecnologia em Saúde
PUCPR**

Rua Imaculada Conceição, 1155, Prado Velho, CEP 80215-901, Curitiba – PR, Brasil
Telefone: +55 41 3271-1657
E-mail: ppgts@pucpr.br

AGRADECIMENTOS

Agradeço, primeiramente, a Deus pela saúde e proteção no dia a dia das minhas atividades.

Agradeço à minha família, ao meu pai, Hélio Gomes de Carvalho, à minha mãe, Lucila Mara Dambiski de Carvalho, e ao meu irmão, Gustavo Dambiski Gomes de Carvalho, que estiveram sempre dispostos a me ajudar e me estimularam para fazer o meu melhor trabalho.

Agradeço, especialmente, ao meu orientador, Professor Dr. Guilherme Nunes Nogueira Neto, por toda a paciência, orientação acadêmica e por sempre estar disposto a auxiliar em todos os momentos. E também ao meu coorientador, Professor Dr. Sergio Ossamu Ioshii, por todo apoio recebido e por acreditar no meu trabalho.

A todos os professores e colegas do Programa de Pós-Graduação em Tecnologia em Saúde (PPGTS), o meu agradecimento pelas diferentes contribuições que deram ao longo da construção deste trabalho.

Agradeço ao Ivan Lucas Reis Silva da Hilab, por toda a ajuda e auxílio nas ideias e aplicações, como também a disponibilidade de tempo para reuniões para cada etapa de evolução no trabalho.

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo apoio financeiro com bolsa taxista durante todo o período de mestrado.

RESUMO

A super-resolução é uma técnica que procura melhorar a resolução das imagens em geral. No caso da área das células sanguíneas, ajuda a identificar e classificar as células corretamente. A fim de obter melhores diagnósticos de doenças, melhorando a análise processo de imagens microscópicas, é necessário, principalmente, a qualidade da imagem para que profissionais de saúde possam chegar a um diagnóstico mais próximo do ideal. Para quem visa implementar algoritmos de super-resolução para a análise de microscopia imagens de células sanguíneas, é útil determinar quais algoritmos estão em uso, para quais propósitos e tendências futuras ou lacunas atuais. Infelizmente não foi encontrado na literatura uma revisão sobre técnicas de super-resolução com foco em sangue células até o momento. Assim, existe o diferencial deste estudo que foi fazer também uma revisão para identificar quais são as técnicas mais utilizadas, por que e para que servem. Além disso, após a análise geral da literatura, o resultado que mais se destacou foi a CNN. Em seguida, foram elaboradas 2 CNNs, a primeira para fazer a super-resolução das imagens e outra para a classificação das células sanguíneas. Depois dos experimentos feitos modificando entre diferentes estruturas para atingir o melhor resultado (em ambas as CNNs), os resultados que foram alcançados nas métricas de classificação foram: na precisão 95,18%, na acurácia 95,19%, na sensibilidade 95,19% e na especificidade 95,19%, sendo resultados ótimos que estão dentro do esperado e que apresentam estabilidade por não terem uma variação alta entre seus valores. Portanto, além dos ótimos resultados alcançados também foi feita uma revisão de literatura auxiliando os pesquisadores a encontrar a melhor técnica para seu estudo, principalmente ser de interesse de pesquisadores e profissionais da patologia, que estão envolvidos no aprimoramento da imagem.

Palavras-chave: Super-resolução. Aprendizado de máquina. Redes neurais. Aprendizado profundo.

ABSTRACT

Super-resolution is a technique that seeks to improve the resolution of images in general. In the case of the blood cell area, it helps to identify and classify the cells correctly. In order to obtain better diagnoses of diseases, improving the analysis process of microscopic images, it is necessary, mainly, the quality of the image so that health professionals can reach a diagnosis closer to the ideal. For anyone looking to implement super-resolution algorithms for the analysis of microscopy images of blood cells, it is useful to determine which algorithms are in use, for what purposes and future trends or current gaps. Unfortunately, a review on super-resolution techniques focused on blood cells was not found in the literature so far. Thus, there is the differential of this study, which was also to carry out a review to identify which are the most used techniques, why and what they are for. In addition, after a general analysis of the literature, the result that stood out the most was the CNN, so 2 CNNs were prepared, the first one for super-resolution of the images and the other for the classification of blood cells. After the experiments done modifying between different structures to achieve the best result (in both CNNs), the results that were achieved in the classification metrics were: in precision 95.18%, in accuracy 95.19%, in sensitivity 95.19 % and 95.19% in specificity, with excellent results that are within the expected range and that are stable because they do not have a high variation between their values. So, in addition to the excellent results achieved, a literature review was also carried out, helping researchers to find the best technique for their study, mainly being of interest to pathology researchers and professionals, who are involved in image enhancement.

Keywords: Super Resolution. Machine Learning. Neural Networks. Deep Learning.

LISTA DE QUADROS

Quadro 1 - Visão Geral dos Artigos da Pesquisa 01	35
Quadro 2 - Destaques da Pesquisa 01	48
Quadro 3 - Visão Geral dos Artigos da Pesquisa 02 da seção Técnicas para Melhorar Imagens Médicas	49
Quadro 4 - Visão Geral dos Artigos da Pesquisa 02 da seção Técnicas para Imagens de Células em Geral.....	51
Quadro 5 - Visão Geral dos Artigos da Pesquisa 02 da Sessão Técnicas para Imagens de RBCs.....	56
Quadro 6 - Visão Geral dos Artigos da Pesquisa 02 da seção Técnicas para Imagens de WBCs	68
Quadro 7 - Visão Geral dos Artigos da Pesquisa 02 da Sessão Técnica para Imagens de RBCs e WBCs.....	86
Quadro 8 - Visão Geral dos Artigos da Pesquisa 02 da Sessão Técnicas para Imagens de Todas as Principais Células Sanguíneas.....	89
Quadro 9 - Destaque da Pesquisa 02.....	94
Quadro 10 - Categorização dos Artigos da Pesquisa 01.....	97
Quadro 11 - Categorização dos Artigos da Pesquisa 02.....	98
Quadro 12 - Comparação entre o trabalho próprio com outros artigos	186

LISTA DE TABELAS

Tabela 1 - - Exemplo de Comparação entre Imagem Original e Imagem de Entrada	117
Tabela 2 - Exemplo de Comparação entre Imagem Original e Imagem de Saída...	118
Tabela 3 - Exemplo de Variância do Laplaciano das Imagens*	118
Tabela 4 - Resultados da Mudança das Camadas	137
Tabela 5 - Resultado Final da Mudança de Camadas	138
Tabela 6 - Resultado Final da Mudança dos <i>Features Maps</i>	139
Tabela 7 - Resultado Final da Mudança do <i>Dropout</i>	140
Tabela 8 - Resultado Final da Mudança das Funções de Ativação	141
Tabela 9 - Resultado Final da Mudança de <i>kernel size e Padding</i>	142
Tabela 10 - Resultado Final da Mudança do <i>Bias</i>	142
Tabela 11 - Resultado Final da Mudança do <i>Kernel Initializer</i>	142
Tabela 12 - Resultado Final da Mudança do Otimizador	143
Tabela 13 - Resultado Final da Mudança do Taxa de Aprendizado	144

LISTA DE FIGURAS

Figura 1 - Comparação entre recortes das Imagens de Microscopia com Resoluções Diferentes.....	28
Figura 2 - Dimensão da Imagem de Microscopia com Resolução Baixa e Alta.....	29
Figura 3 - Estrutura Geral	33
Figura 4 - Diagrama PRISMA da pesquisa 01	97
Figura 5 - Diagrama PRISMA da Pesquisa 02	99
Figura 6 - Esquema Geral do Método Proposto.....	100
Figura 7 - Algoritmo YOLO	101
Figura 8 - Exemplo de Módulo de Inception com dimensões reduzidas.....	101
Figura 9 - Algoritmo VGG16	102
Figura 10 - Divisão do Código no Google Colab.....	103
Figura 11 - Separação das imagens	104
Figura 12 - Separação das imagens - Fluxograma	105
Figura 13 - DownSample ou Resize	105
Figura 14 - DownSample ou Resize - Fluxograma.....	106
Figura 15 - Cropped - Processo 1.....	107
Figura 16 - Cropped - Fluxograma 1.....	108
Figura 17 - Cropped - Processo 2.....	108
Figura 18 - Cropped - Fluxograma 2.....	109
Figura 19 - Data Augmentation - Processo	110
Figura 20 - Data Augmentation - Fluxograma	111
Figura 21 - Data Augmentation - Divisão	112
Figura 22 - Super Resolução - Tempo Esgotado	112
Figura 23 - Super Resolução - Fluxograma Jupyter.....	113
Figura 24 - Super Resolução - Fluxograma Possibilidades.....	114
Figura 25 - Super Resolução - Fluxograma Finalização	115
Figura 26 - Super Resolução - Mudança - Mudanças das Camadas	122
Figura 27 - Super Resolução - Mudança - Filtros.....	122
Figura 28 - Super Resolução - Mudança - <i>Dropout</i>	123
Figura 29 - Exemplo de Recolocação do 13 x 13 nas Camadas 1 e 2	123
Figura 30 - Exemplo de Distribuição do 13 x 13 nas Camadas 2 e 3.....	124

Figura 31 - Modelos com os valores de Bias com True e False	124
Figura 32 - Exemplos de modelos para gerar imagens com Super Resolução	125
Figura 33 - Exemplo de Matriz de Confusão	128
Figura 34 - Exemplos de Gráficos de Aprendizagem de Acurácia e de Perda.....	129
Figura 35 - Divisão do <i>dataset</i>	130
Figura 36 - Imagens de LR de diferentes tipos de células	131
Figura 37 - Imagens de SR (4x) de diferentes tipos de células.....	131
Figura 38 - Primeira mudança de substituição do <i>dataset</i>	132
Figura 39 - Segunda mudança de substituição do <i>dataset</i>	132
Figura 40 - Terceira mudança de substituição do <i>dataset</i>	133
Figura 41 - Quarta mudança de substituição do <i>dataset</i>	133
Figura 42 - Quinta mudança de substituição do <i>dataset</i>	134
Figura 43 - Primeira, segunda e terceira mudanças de adição do <i>dataset</i>	135
Figura 44 - Quarta mudança de adição do <i>dataset</i>	136
Figura 45 - Quinta mudança de adição do <i>dataset</i>	136
Figura 46 - Melhor Modelo com Mudanças de Camadas	138
Figura 47 - Melhor Modelo com Mudanças dos <i>Features Maps</i>	139
Figura 48 - Exemplo de <i>Frame</i> na Imagem de Saída com a mudança do <i>Dropout</i> .140	
Figura 49 - Melhor Modelo com Mudanças do <i>Dropout</i>	141
Figura 50 - Exemplo de <i>Frame</i> na Imagem de Saída com Otimizador SGD	143
Figura 51 - Exemplo de <i>Frame</i> na Imagem de Saída com diferentes <i>Learning Rates</i>	144
Figura 52 - Modelo Final “Vencedor”	144
Figura 53 - Imagens de Entrada para Teste com diferentes Escalas.....	145
Figura 54 - Imagens de Saída para Teste com diferentes Escalas.....	145
Figura 55 - Super Resolução - Gráficos Resultados PSNR	146
Figura 56 - Super Resolução - Gráficos Resultados SSIM	147
Figura 57 - Super Resolução - Gráficos Resultados Variância do Laplaciano	147
Figura 58 - Resultados do Teste da CNN com Imagens Originais e com Imagens de SR.....	148
Figura 59 - Resultados do Teste da SVM com Imagens Originais e com Imagens de SR.....	148
Figura 60 - Gráficos de Aprendizado e Valores por Época – 100% das Imagens Originais.....	149

Figura 61 - Matriz de Confusão – 100% das Imagens Originais	149
Figura 62 - Relatório de Classificação e Métricas de Classificação – 100% das Imagens Originais	150
Figura 63 - Classificação das Imagens de LR – 100% das Imagens Originais.....	151
Figura 64 - Classificação das Imagens de SR – 100% das Imagens Originais	152
Figura 65 - Gráficos de Aprendizado e Valores por Época – 100% das Imagens de SR	152
Figura 66 - Matriz de Confusão – 100% das Imagens de SR	153
Figura 67 - Relatório de Classificação e Métricas de Classificação – 100% das Imagens de SR.....	153
Figura 68 - Classificação das Imagens de LR – 100% das Imagens de SR.....	154
Figura 69 - Classificação das Imagens de SR – 100% das Imagens de SR	155
Figura 70 - Gráficos de Aprendizado e Valores por Época – Primeira mudança de Substituição.....	156
Figura 71 - Gráficos de Aprendizado e Valores por Época – Segunda mudança de Substituição.....	156
Figura 72 - Gráficos de Aprendizado e Valores por Época – Terceira mudança de Substituição.....	156
Figura 73 - Matriz de Confusão – Primeira, Segunda e Terceira mudanças de Substituição.....	157
Figura 74 - Relatório de Classificação e Métricas de Classificação – Primeira mudança de Substituição.....	158
Figura 75 - Relatório de Classificação e Métricas de Classificação – Segunda mudança de Substituição.....	158
Figura 76 - Relatório de Classificação e Métricas de Classificação – Terceira mudança de Substituição.....	158
Figura 77 - Classificação das Imagens de LR – Primeira mudança de Substituição	159
Figura 78 - Classificação das Imagens de LR – Segunda mudança de Substituição	160
Figura 79 - Classificação das Imagens de LR – Terceira mudança de Substituição	161
Figura 80 - Classificação das Imagens de SR – Primeira mudança de Substituição	161

Figura 81 - Classificação das Imagens de SR – Segunda mudança de Substituição	162
Figura 82 - Classificação das Imagens de SR – Terceira mudança de Substituição	163
Figura 83 - Gráficos de Aprendizado e Valores por Época – Quarta mudança de Substituição	164
Figura 84 - Gráficos de Aprendizado e Valores por Época – Quinta mudança de Substituição	165
Figura 85 - Matriz de Confusão – Quarta e Quinta mudanças de Substituição.....	166
Figura 86 - Relatório de Classificação e Métricas de Classificação – Quarta mudança de Substituição	166
Figura 87 - Relatório de Classificação e Métricas de Classificação – Quinta mudança de Substituição	167
Figura 88 - Classificação das Imagens de LR – Quarta mudança de Substituição .	167
Figura 89 - Classificação das Imagens de LR – Quinta mudança de Substituição .	168
Figura 90 - Classificação das Imagens de SR – Quarta mudança de Substituição .	169
Figura 91 - Classificação das Imagens de SR – Quinta mudança de Substituição .	170
Figura 92 - Gráficos de Aprendizado e Valores por Época – Primeira, Segunda e Terceira mudanças de Adição.....	172
Figura 93 - Matriz de Confusão – Primeira, Segunda e Terceira mudanças de Adição	173
Figura 94 - Relatório de Classificação e Métricas de Classificação – Primeira, Segunda e Terceira mudanças de Adição.....	174
Figura 95 - Classificação das Imagens de LR – Primeira mudança de Adição	175
Figura 96 - Classificação das Imagens de LR – Segunda mudança de Adição	176
Figura 97 - Classificação das Imagens de LR – Terceira mudança de Adição	176
Figura 98 - Classificação das Imagens de SR – Primeira mudança de Adição.....	177
Figura 99 - Classificação das Imagens de SR – Segunda mudança de Adição.....	178
Figura 100 - Classificação das Imagens de SR – Terceira mudança de Adição	179
Figura 101 - Gráficos de Aprendizado e Valores por Época – Quarta e Quinta mudanças de Adição.....	180
Figura 102 - Matriz de Confusão – Quarta e Quinta mudanças de Adição	181
Figura 103 - Relatório de Classificação e Métricas de Classificação – Quarta e Quinta mudanças de Adição.....	181

Figura 104 - Classificação das Imagens de LR – Quarta mudança de Adição	182
Figura 105 - Classificação das Imagens de LR – Quinta mudança de Adição	183
Figura 106 - Classificação das Imagens de SR – Quarta mudança de Adição	184
Figura 107 - Classificação das Imagens de SR – Quinta mudança de Adição	184

LISTA DE ABREVIATURAS E SIGLAS

ABC	<i>Artificial Bee Colony</i>
AlexNet	<i>ImageNet Classification with Deep Convolutional Neural Networks</i>
ALO	<i>Ant Lion Optimizer</i>
ANN	<i>Artificial Neural Network</i>
AP	<i>Average Precision</i>
BA	<i>Bat Algorithm</i>
BPNN	<i>Back Propagation Neural Networks</i>
BSA-Net	<i>Bilinear Self-Attention Network</i>
BSV	<i>Boundary Support Vectors</i>
CBC	<i>Complete Blood Cell Counts</i>
CEUS	<i>Contrast-Enhanced Ultrasound</i>
CHT	<i>Circular Hough Transform</i>
CIS	<i>CMOS Image Sensor</i>
CMOS	<i>Complementary Metal-Oxide-Semiconductor</i>
CNN	<i>Convolutional Neural Network</i>
CNN-SR	<i>Convolutional Neural Network Based Super Resolution</i>
COFA	<i>Chaotic Optimal Foraging Algorithm</i>
CR	<i>Compression Rate</i>
CSRCM	<i>Cascade Sparse Regression Chain Model</i>
CSRNet	<i>Cell Super Resolution Network</i>
CSRS	<i>Classification-Segmentation Reversible System</i>
dCNN	<i>Deep Convolutional Neural Network</i>
Densenet	<i>Densely Connected Convolution Network</i>
DLSF	<i>Direct Least Squares Fitting</i>
DNN	<i>Deep Neural Network</i>
DT	<i>Decision Tree</i>
dU-Net	<i>Deformable U-Net</i>
ELMSR	<i>Extreme Learning Machine-Based Super Resolution</i>
ESA	<i>Evidential Segmentation Algorithm</i>
FCM	<i>Fuzzy C Mean</i>
FCN	<i>Fully Connected Network</i>

FCNN	<i>Fuzzy Cellular Neural Network</i>
FCT	<i>Fast Circlet Transform</i>
FDCT	<i>Forward Discrete Curvelet Transform</i>
FDDE	<i>Frequency-Domain Diagonal Extension</i>
FED	<i>Fast and Efficient YOLOv3</i>
FNR	<i>False Negative Rate</i>
fPALM	<i>Fluorescence Photo-Activated Localization Microscopy</i>
FPM	<i>Fourier Ptychographic Microscopy</i>
FSRCNN	<i>Fast Convolutional Neural Network Super Resolution</i>
GAN	<i>Generative Adversarial Network</i>
GFS	<i>Geometric-Feature Spectrum ExtremeNet</i>
GLCM	<i>Gray Level Co-occurrence Matrix</i>
GLRLM	<i>Gray Level Run Length Matrix</i>
GoogleNet	<i>Going Deeper with Convolutions</i>
GVF	<i>Gradient Vector Flow</i>
HE	<i>Histogram Equalization</i>
HSVM	<i>Hierarchical SVM</i>
IO	<i>Iterative Otsu</i>
IRHT	<i>Iterative Random Hough Transformation</i>
KNCN	<i>K-Nearest Centroid Neighbors</i>
KNN	<i>K-Nearest Neighbors</i>
L-PAM	<i>Localization Photoacoustic Microscopy</i>
L-PAM-GS	<i>Localization Photoacoustic Microscopy System with a Galvanometer Scanner</i>
LED	<i>Light-Emitting Diode</i>
MAWA	<i>Maximum Adaptive Weighted Averaging Filter</i>
MB	<i>Microbubbles</i>
MFO	<i>Moth Flame Optimization</i>
ML	<i>Machine Learning</i>
MLP	<i>Multilayer Perceptron</i>
MRA	<i>Magnetic Resonance Angiography</i>
mULM	<i>Model Ultrasound Localization Microscopy</i>
MVO	<i>Multiverse Optimization</i>

NA	<i>Numerical Apertures</i>
NDA	<i>New Detection Algorithm</i>
NN	<i>Neural Network</i>
NSS	<i>Neutrosophic Similarity Score</i>
OCT	<i>Optical Coherence Tomography</i>
OFA	<i>Optimal Foraging Algorithm</i>
OMAG	<i>Optical Microangiography</i>
PA	<i>Photoacoustic</i>
PAI	<i>Photoacoustic Imaging</i>
PAM	<i>Photoacoustic Microscopy</i>
PCA	<i>Principal Component Analysis</i>
POC	<i>Point of Care</i>
PPED	<i>Projected Principal-Edge Distribution</i>
RANSAC	<i>Random Sample Consensus</i>
RBC	<i>Red Blood Cell</i>
RBF	<i>Radial Basis Functions</i>
rBV	<i>Relative Blood Volume</i>
RCF	<i>Radial-Based Cell Formation</i>
RCNN	<i>Region Convolutional Neural Network</i>
REFPM	<i>Resolution-Enhanced Fourier Ptychographic Microscopy</i>
ResNet	<i>Residual Neural Network</i>
ROC	<i>Receiver Operating Characteristic Curves</i>
ROI	<i>Region of Interest</i>
RPN	<i>Region Proposal Networks</i>
rVTT	<i>Relativistic Visual Turing Test</i>
SBP	<i>Space-Bandwidth-Product</i>
SF	<i>Source Follower</i>
SFS	<i>Sequential Forward Selection</i>
SIM	<i>Structured Illumination Microscopy</i>
SISR	<i>Single Image Super Resolution</i>
SNR	<i>Signal-to-Noise Ratio</i>
SR	<i>Super Resolution</i>
SRM	<i>Super-Resolution Microscopy</i>

SRNet	<i>Super-Resolution Network</i>
SROFM	<i>Sub-Pixel Resolving Optofluidic Microscope</i>
SSD	<i>Single Shot MultiBox Detector</i>
STED	<i>Stimulated Emission Depletion</i>
STORM	<i>Stochastic Optical Reconstruction Microscopy</i>
SUSHI	<i>Sparsity-based Ultrasound Super-resolution Hemodynamic Imaging</i>
SVM	<i>Support Vector Machine</i>
TSMM	<i>Threshold Segmentation followed by Mathematical Morphology</i>
TWO-DCNN	<i>Two-module Weighted Optimized Deformable Convolutional Neural Networks</i>
U-Net	<i>Universal Network</i>
ULM	<i>Ultrasound Localization Microscopy</i>
UV	<i>Ultraviolet</i>
VGG	<i>Very Deep Convolutional Networks for Large-Scale Image Recognition</i>
WBC	<i>White Blood Cells</i>
WSSA	<i>Wavelet-based Segmentation for Spherical Images</i>
YOLO	<i>You Only Look Once</i>

SUMÁRIO

1	INTRODUÇÃO	27
1.1	APRESENTAÇÃO, PROBLEMATIZAÇÃO E MOTIVAÇÃO	27
1.2	OBJETIVO GERAL E OBJETIVOS ESPECÍFICOS.....	31
1.2.1	Objetivo Geral	31
1.2.2	Objetivo Específicos	32
1.3	JUSTIFICATIVA	32
1.3.1	Justificativa Teórica.....	32
1.3.2	Justificativa Empírica	32
1.4	HIPÓTESE	33
1.5	VISÃO GERAL DA DISSERTAÇÃO	33
2	FUNDAMENTAÇÃO TEÓRICA	34
2.1	PESQUISA 01.....	34
2.1.1	Técnicas Genéricas	35
2.1.2	Técnicas para Imagens Vasculares	39
2.1.3	Técnicas para Imagens Celulares	42
2.1.4	Técnicas para Imagens de Células Sanguíneas	43
2.2	PESQUISA 02.....	49
2.2.1	Técnicas para Melhorar Imagens Médicas	49
2.2.2	Técnicas para Imagens de Células em Geral	51
2.2.3	Técnicas para Imagens de RBCs	55
2.2.4	Técnicas para Imagens de WBCs	67
2.2.5	Técnicas para Imagens de RBCs e WBCs.....	85
2.2.6	Técnicas para Imagens das Principais Células Sanguíneas.....	89
3	MATERIAIS E MÉTODOS	96
3.1	MATERIAIS.....	96

3.2	CLASSIFICAÇÃO E DELINEAMENTO DO ESTUDO	96
3.3	METODOLOGIA.....	96
3.4	MÉTODO PROPOSTO.....	99
4	RESULTADOS.....	137
4.1	RESULTADOS ALCANÇADOS	137
5	DISCUSSÃO	187
6	CONCLUSÃO	189
6.1	CONTRIBUIÇÃO CIENTÍFICA, TECNOLÓGICA E SOCIAL.....	189
6.2	TRABALHOS FUTUROS.....	190
	REFERÊNCIAS.....	191

1 INTRODUÇÃO

Esta seção inicial está dividida em cinco seções: Problematização e Motivação, Objetivo Geral e Objetivos Específicos, Justificativa, Hipótese e Visão Geral da Dissertação.

1.1 APRESENTAÇÃO, PROBLEMATIZAÇÃO E MOTIVAÇÃO

A área da saúde possui diversos estudos importantes. Um deles pesquisa sobre as células do sangue e utiliza equipamentos de última geração para conseguir um diagnóstico mais próximo ao ideal.

O sangue é composto por quatro partes. A primeira é composta de células, iniciando-se pelas células vermelhas (RBCs) ou hemácias, que são responsáveis pelo transporte do oxigênio no organismo. Seguindo, existem também as células brancas (WBCs) ou leucócitos, que são fundamentais para a defesa do corpo. Como também existem as plaquetas, as quais são encarregadas de atuar na coagulação sanguínea. E o último componente é o plasma que é o líquido em que as células se encontram mergulhadas (ZHANG *et al.*, 2019; DESHPANDE; GITE; ALUVALU, 2021; TESSEMA *et al.*, 2021).

O funcionamento correto das células de sangue é vital para a vida, por isso o exame de sangue é uma ferramenta importante para encontrar diversas doenças que podem assolar os seres humanos (DESHPANDE; GITE; ALUVALU, 2021).

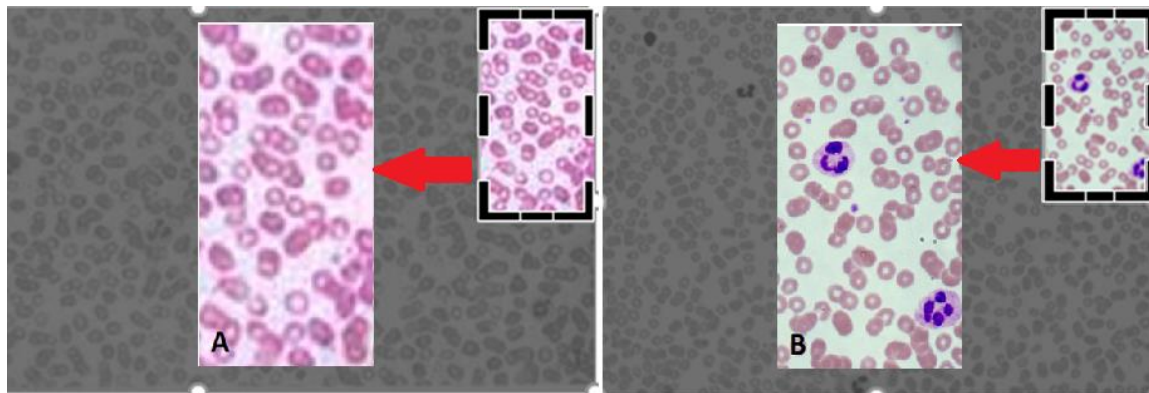
O trabalho de identificação e classificação de células é árduo, trabalhoso e depende de um profissional de saúde habilitado, pois além de ser manual, requisita diversas horas para que o médico que estuda os fluidos do corpo humano, conhecido como patologista, mais especificamente hematopatologista (pois o estudo é focado no sangue), identifique e classifique cada célula, observando se existe alguma anormalidade para que seja possível reconhecer as doenças que possam ocorrer (ALAM; ISLAM, 2019).

Existe, ainda, um agravante, onde a identificação e classificação das imagens é mais complicada quando a resolução das imagens é menor do que ideal (WANG; CHEN; HOI, 2021).

Por exemplo, a Figura 01 mostra a imagem de células de sangue com resoluções diferentes, demonstrando também recorte e aplicando um “zoom” para

facilitar a comparação. Assim, é possível observar, visualmente, a resolução diferente das imagens. Logo, para ser possível identificar e classificar de forma mais precisa, é preciso que as imagens tenham uma boa resolução (WANG; CHEN; HOI, 2021).

Figura 1 - Comparação entre recortes das Imagens de Microscopia com Resoluções Diferentes



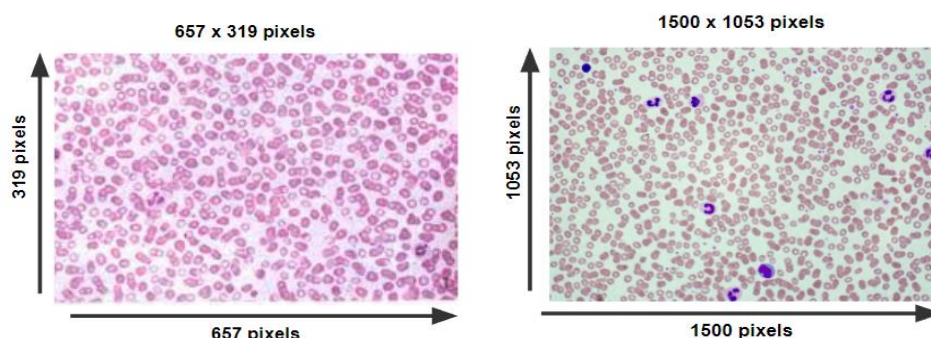
A - Imagem com menor resolução / B - Imagem com maior resolução

Fonte: autoria própria.

Após o recorte, como mostra a Figura 1, é possível observar que a imagem “A” está “borrada” ou desfocada e “B” está com as bordas definidas, permitindo distinguir de forma mais fácil as células que existem na imagem.

Assim, as imagens são formadas por pixels que definem sua resolução (EL-SHAFAI *et al.*, 2022). A imagem com baixa resolução possui 319 pixels de altura e 657 pixels de comprimento e a imagem com alta resolução com 1.053 de altura e 1.500 pixels de comprimento (vide Figura 2). Desse modo, como a imagem de alta resolução possui mais pixels, ela possui mais detalhes a serem mostrados. Logo, a resolução é melhor, o que implica na detecção melhor de cada objeto dentro da imagem. Já a de baixa resolução é mais propícia a ter erros, sendo que não é possível observar de forma clara cada célula (CLARO *et al.*, 2020).

Figura 2 - Dimensão da Imagem de Microscopia com Resolução Baixa e Alta



Fonte: autoria própria.

Nesse contexto, para se atingir o mais próximo em diagnosticar mais precisamente, os laboratórios, *startups* e outras empresas da área da saúde investem na busca de tecnologias que desenvolvam equipamentos e sistemas (*hardware* e *software*) de ponta para que seja possível melhorar a resolução e a classificação das imagens e entregar um diagnóstico mais preciso e mais rápido. Entretanto, o ideal é que haja um bom custo-benefício entre ser extremamente tecnológico e que não tenha um valor inacessível de aquisição e manutenção. Com isso, a procura por inteligência artificial aumentou, pois esta oferece um trabalho automatizado, podendo receber um elevado volume de análises, diminuindo os erros e com precisão alta, além de ser realizada em pouco tempo em comparação com o profissional de saúde, reduzindo, também, os custos (KHASHMAN, 2012; KASAMSUMRAN *et al.*, 2020; LONG *et al.*, 2021).

Desta forma, existem diversas técnicas que focam em melhorar a resolução de imagens microscópicas, conhecidas como métodos de super-resolução, existindo técnicas genéricas, como as que usam equações de Fourier (LI *et al.*, 2006; SUN *et al.*, 2017). Também existem métodos com destaque para imagens de microvasos (CAI *et al.*, 2011) ou fluxo sanguíneo (OPACIC *et al.*, 2018) e, principalmente, metodologias para células (WESTMORELAND *et al.*, 2016), especialmente em células sanguíneas (ZHENG *et al.*, 2011; BHANDARI *et al.*, 2017; LIU *et al.*, 2017).

Em relação à identificação e classificação de imagens microscópicas de células, existem algoritmos automatizados com base em inteligência artificial para realizar o trabalho de auxiliar o profissional da saúde no diagnóstico de doenças, visando alcançar o melhor tratamento possível. Nesse sentido, há métodos que focam

em reconhecer células em geral, como verificar se os rótulos de dados das células estão corretos (LIU *et al.*, 2020). Há outros para distinguir células sanguíneas que estão sobrepostas, aglomeradas ou que se encostam (SONG; WANG, 2009) e também aquele que detecta e reconstrói os contornos dos objetos desejados, inclusive podendo ser células sanguíneas (NILUFAR; PERKINS, 2014). Existem outros que destacam algumas ou todas as principais células do sangue (SHAHZAD *et al.*, 2020), que são as hemácias (FREJLICHOWSKI, 2012), os leucócitos (MORADIAMIN *et al.*, 2016) e as plaquetas (WESTMORELAND *et al.*, 2016). Existem desafios tanto para melhorar a resolução quanto para identificar e categorizar as imagens microscópicas de células de sangue, como células aglomeradas ou sobrepostas, células anormais, células de formatos diferentes por causa de alguma doença, células incompletas etc. Para alcançar os objetivos de melhorar a resolução e de identificar e classificar as imagens, além de conseguir ter a possibilidade de superar esses desafios, o aprendizado de máquina (*machine learning*), mais especificamente o aprendizado profundo (*deep learning*), focando em redes neurais convolucionais (CNN), é a chave do processo. De acordo com Lu *et al.* (2017), as redes neurais profundas (do aprendizado profundo) são, normalmente, utilizadas para analisar as imagens de microscopia, como classificadores ou extratores de características, com o objetivo de detectar, segmentar e classificar os objetos pedidos. As CNNs são, de acordo com esse mesmo autor, as redes mais populares para a aplicação de análise de imagem microscópica, sendo que, se for para o uso de extração de recursos, gera-se uma rede que é uma representação transformada de cada imagem de entrada, podendo ser aplicada à análise de dados, como, por exemplo, seleção de recursos ou classificação de alvos.

Existem diversos tipos de CNNs, sendo as mais conhecidas: YOLO (que faz detecção e classificação de objetos), Inception e VGG16 (somente fazem a classificação da imagem como um todo). Segundo Redmon *et al.* (2016) e a revisão de Oliveira *et al.* (2021), a YOLO é uma rede com base em regressão que usa um único *backbone* de CNN para prever caixas limitadoras que possuem rótulos com as probabilidades em um único esquema de avaliação. Já para a Inception, de acordo com Szegedy *et al.* (2015) e a revisão de Oliveira *et al.* (2021), propõe-se o conceito de módulos de iniciação, com o objetivo de ser usado para focar na extração massiva de características locais. Essa rede de arquitetura teve como intuito alcançar o mesmo desempenho de camadas empilhadas sequencialmente, sendo que treina ao mesmo

tempo várias camadas convolucionais e vai “empilhando” seus mapas de características vinculados a um perceptron de multicamadas. Finalmente, segundo Simonyan e Zisserman (2014) e a revisão de Oliveira *et al.* (2021), o modelo VGG16 é usado para classificação de imagens, composto por dezesseis camadas convolucionais, múltiplas camadas *max-pool* e três camadas finais totalmente conectadas.

Para avaliar se os algoritmos estão funcionando da melhor forma possível, existem métricas para verificar se atuam do modo correto, sendo as mais comuns: sensibilidade, especificidade, precisão média e acurácia. Alguns exemplos, como no artigo de Jiao *et al.* (2018), os valores obtidos pelo algoritmo CNN proposto foram 99% em precisão, 98,4% em sensibilidade e 99,3% em acurácia. Já no artigo de Grochowski *et al.* (2019), o algoritmo atingiu os resultados nas métricas de acurácia, de sensibilidade e de especificidade de 94,56%, 89,13% e 100%, respectivamente. Já o artigo de Wang *et al.* (2019), além de registrar outras métricas, focou nas mais utilizadas que alcançaram 90,85% na acurácia e 94,69% na precisão. Outro exemplo é o de Malkawi *et al.* (2020) que demonstraram que o método proposto atingiu 98,7% em acurácia, 98,6% em sensibilidade, 99,7% em especificidade e 98,8% em precisão. Por fim, Mohamed *et al.* (2020) compararam diferentes algoritmos, sendo que o que obteve melhor resultado (DenseNet-169) atingiu a precisão de 86%, a sensibilidade de 86%, a especificidade de 95% e a acurácia de 92%. Pode-se observar que os valores são altos e precisos nesses exemplos.

Por isso, a procura por algoritmos com diferentes técnicas que fizessem o reconhecimento automático, com mais precisão e menor tempo de operação aumentou muito com o passar dos anos, melhorando a precisão dos diagnósticos das doenças, para ter um tratamento mais próximo ao ideal, ajudando na vida das pessoas (COOKE *et al.*, 2021). E, por apresentar diversas técnicas e ser uma área recente, necessita de mais estudos aprofundados.

1.2 OBJETIVO GERAL E OBJETIVOS ESPECÍFICOS

1.2.1 Objetivo Geral

Investigar técnicas de super-resolução aplicadas à classificação de imagens microscópicas de células sanguíneas humanas, visando analisar se existe o impacto positivo nas métricas de classificação.

1.2.2 Objetivo Específicos

São objetivos específicos deste trabalho:

- Revisar a Literatura sobre as técnicas de melhoria da resolução de imagens.
- Selecionar técnicas e algoritmos apropriados para a identificação e a classificação de células sanguíneas no escopo de melhoria de resolução.
- Implementar algoritmos com ajustes voltados à melhoria da resolução de imagens.
- Comparar desempenho de algoritmos de identificação e classificação de células sanguíneas visando a obtenção das melhores métricas.

1.3 JUSTIFICATIVA

1.3.1 Justificativa Teórica

Existem estudos que buscam classificar as células sanguíneas por meio de algoritmos de inteligência artificial (SHAHZAD *et al.*, 2020). Contudo, considerando a limitação de uma resolução mais baixa e a dificuldade de alcançar uma melhor resolução das células por meio de algoritmos, mais estudos aprofundando as técnicas são necessários nesse contexto (MAISELI *et al.*, 2014).

1.3.2 Justificativa Empírica

O algoritmo desenvolvido poderá contribuir em exames clínicos para serem mais viáveis, práticos e rápidos. Inclusive, por exemplo, terá potencial de ser usado pela empresa Hilab em conjunto com seus equipamentos que reconhecem e categorizam as células sanguíneas.

1.4 HIPÓTESE

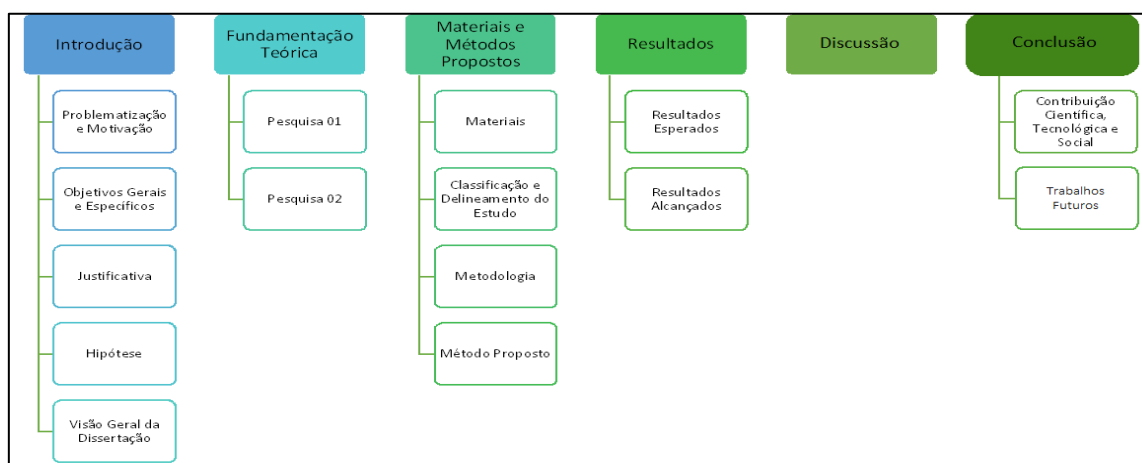
Em alinhamento com os estudos recentes citados anteriormente, serão utilizadas as quatro métricas mais comuns para classificação: precisão, acurácia, sensibilidade e especificidade (como destacado na seção 1.1 Apresentação, Problematização e Motivação). O foco está em observar se ocorre a contribuição e impacto positivo com a implementação da super-resolução.

1.5 VISÃO GERAL DA DISSERTAÇÃO

Este trabalho está dividido em cinco seções: Introdução, Fundamentação Teórica, Materiais e Métodos Propostos, Resultados e Contribuições, e Conclusão.

A Figura 3 exibe a estrutura geral.

Figura 3 - Estrutura Geral



Fonte: autoria própria.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta os artigos recolhidos das bases, sendo a Pesquisa 01 mais geral para técnicas de super-resolução e a Pesquisa 02 para focar em métodos que identificam e classificam as células sanguíneas. A seção está dividida em Pesquisa 01 e Pesquisa 02 com suas subseções dentro de cada pesquisa correspondente. A metodologia para a seleção dos artigos está descrita na seção 3 “Materiais e Métodos”.

Dentro das seções “Pesquisa 01” e “Pesquisa 02” existem subseções, sendo que na introdução de cada um deles, existe uma apresentação geral de cada artigo abordado. Logo após, dentro das subseções têm-se as resenhas de cada artigo e os propósitos, além da essência do trabalho, finalizando com a indicação das lacunas identificadas.

2.1 PESQUISA 01

A Pesquisa 01 teve como objetivo encontrar técnicas de super-resolução em imagens microscópicas do sangue, criando um embasamento de métodos existentes para melhorar a resolução em imagens. Com essa pesquisa, alcançou-se 223 artigos, sendo dela retirados 191 por estarem diferente do escopo. Assim, foi atingido um total de 32 artigos que estão dentro ou tangentes ao escopo do tema do trabalho.

O Quadro 1 apresenta um resumo da Pesquisa 01 e de quais as técnicas usadas pelos autores. Começa identificando o autor, o tema ou categoria que a técnica se encaixa, o nome da técnica e se apresenta detalhamento das técnicas. Mais especificamente, expõem-se os algoritmos utilizados e se evidencia o seu uso em aplicações microscópicas para células sanguíneas. Assim, foram divididos em quatro tópicos, o primeiro é “Genéricos” com 25% dos artigos da Pesquisa 01, o segundo é “Imagens Vasculares” com 28,1% dos artigos, o terceiro é “Imagens Celulares” com 9,4% da pesquisa e o último é “Imagens de Células Sanguíneas” com 37,5% do total. Tais artigos são detalhados nos tópicos do referencial teórico a seguir.

Quadro 1 - Visão Geral dos Artigos da Pesquisa 01

Artigo	Tema/Categoria	Técnicas	Detalhamento das técnicas (Algoritmos)	Aplicações microscópicas para células sanguíneas
Li <i>et al.</i> (2006)	Fourier	MAWA		X
Sun <i>et al.</i> (2017)	Fourier	FPM		
Mudanyali, Bishara e Ozcan (2011)	Holográfico	-		X
Greenbaum <i>et al.</i> (2013)	Holográfico	-	X	
York <i>et al.</i> (2013)	3D	SIM & STED		
Becatti <i>et al.</i> (2020)	Células Anômalas	STED		
Cai <i>et al.</i> (2020)	Esferas	WSSA	X	
Zalevsky <i>et al.</i> (2011)	Super-resolução Geral	-	X	
Lavina e Gaengel (2015)	Geral Vascular	-		
Cai <i>et al.</i> (2011)	Vascular	MRA	X	
Piepenbrock <i>et al.</i> (2019)	Vascular/Ultrassom	SUSHI	X	
Yousefi, Qin e Wang (2013)	Vascular Animal	OMAG	X	
Ackermann e Schmitz (2015)	Vascular	-	X	
Dencks <i>et al.</i> (2018)	Vascular	ULM		
Kim <i>et al.</i> (2019b)	Vascular	PAM		
Opacic <i>et al.</i> (2018)	Vascular	mULM		
Kohler <i>et al.</i> (2014)	Vascular	-	X	
Westmoreland <i>et al.</i> (2016)	Plaquetas	SIM & STORM		
Pennanen <i>et al.</i> (2017)	Células (Não Sangue)	STED		
Jiang <i>et al.</i> (2020b)	Sangue - Domínio da frequência	SIM	X	X
Kim <i>et al.</i> (2019a)	Sangue	PAM	X	
Luo <i>et al.</i> (2016)	Sangue	-	X	X
Bhandari <i>et al.</i> (2017)	Sangue	-	X	X
Huang <i>et al.</i> (2016a)	Sangue	CNN	X	X
Liu <i>et al.</i> (2017)	Sangue	CNN	X	X
Li <i>et al.</i> (2021b)	Sangue	-	X	X
Zheng <i>et al.</i> (2011)	Sangue	SROFM	X	X
Ayas e Ekinici (2020)	Sangue	CNN & SISR	X	X
Huang <i>et al.</i> (2016b)	Sangue	CNNSR & ELMSR	X	X
Maiseli <i>et al.</i> (2014a)	Sangue	-	X	X
Ma <i>et al.</i> (2018)	Sangue	CSRNet & FSRCNN	X	X
Tom <i>et al.</i> (2019)	Sangue	SISR & CNN	X	X

Fonte: autoria própria.

2.1.1 Técnicas Genéricas

Foram identificados oito trabalhos que utilizaram técnicas genéricas no tema da microscopia. Li *et al.* (2006) e Sun *et al.* (2017) usaram diferentes abordagens com as transformadas de Fourier, enquanto Mudanyali, Bishara e Ozcan (2011) e Greenbaum *et al.* (2013) aplicaram métodos com holografia. Por fim, outros quatro

trabalhos empregaram técnicas diversas com o objetivo de melhorar a resolução em diferentes aplicações como: imagens 3D para hemodinâmica (YORK *et al.*, 2013); células anômalas (BECATTI *et al.*, 2020); formato esférico para a retina (CAI *et al.*, 2020); e máscara binária para imagens em geral (ZALEVSKY *et al.*, 2011).

Li *et al.* (2006) desenvolveram uma técnica de análise para imagens dinâmicas para melhorar a resolução das bordas das células, sendo capaz de realizar a medição em tempo real da elasticidade da membrana das RBCs, sendo estimada a flutuação minuto da membrana com base na super-resolução. Assim, a resolução da borda da célula é independente da resolução óptica do microscópio de luz, sendo definido pelo ruído da imagem, o qual foi utilizado um filtro de Média Ponderada Adaptativa Máxima (MAWA) para diminuir o desvio padrão do ruído e com um algoritmo de super-resolução para conseguir uma imagem de alta resolução, através de uma única imagem de baixa resolução. Os autores relatam que utilizando o método, conseguiram medir os parâmetros de hemácias (RBCs) em diferentes condições. No artigo, existem equações, imagens e gráficos que fundamentam os resultados alcançados, porém não é especificado qual e como é usado o algoritmo de super-resolução.

Sun *et al.* (2017) relataram a utilização da técnica de Fourier Ptychographic Microscopy (FPM) aprimorada de resolução baseada em iluminação de alto de aberturas numéricas (NA), chamada de Resolution-Enhanced Fourier Ptychographic Microscopy (REFPM), para gerar reconstruções grandes que são o produto de largura de banda de espaço (SBP) de alta resolução a partir de uma série de imagens brutas de baixa resolução. É demonstrado que o sistema REFPM pode melhorar significativamente a resolução e alcançar o desempenho eficiente de imagem final. No artigo, mostraram imagens (não de sangue) com resolução melhorada, porém não existe aprofundamento em algoritmos ou códigos de super-resolução.

Ambos os artigos não se aprofundaram nos algoritmos de super-resolução utilizados, mas nas técnicas com as equações de Fourier. De modo geral, seus resultados comprovaram que houve melhoria na qualidade das imagens (LI *et al.*, 2006; SUN *et al.*, 2017).

Mudanyali, Bishara e Ozcan (2011) demonstraram a "microscopia de super-resolução de pixel sem lente portátil como uma ferramenta econômica que se baseia na recuperação de digital em linha de alta resolução de hologramas dos objetos através da aquisição de múltiplas armações sem lentes sinterizadas espacialmente".

No artigo, essa técnica é feita usando um algoritmo de super-resolução de pixel iterativo, sendo que esses hologramas sem lentes capturados sequencialmente são colocados digitalmente juntos, recuperando um holograma de objeto de resolução mais alta. Assim, além da técnica, é utilizado como auxílio de um filme umectante que melhoram os resultados experimentais em relação sinal-ruído (SNR) de até 4 vezes, mostrando uma melhor recuperação de características submicrônicas de espécimes, sendo que o filme umectante permite um efeito estável e repetível de microlentes para aprimorar as capacidades do holográfico sem lentes portáteis em campo microscópios. Além de toda a explicação sobre a técnica no artigo, existem imagens que demonstram a alta resolução e que pode identificar RBCs e alguns outros tipos de células, porém o artigo não foca em algoritmos ou códigos de super-resolução.

Greenbaum *et al.* (2013) desenvolveram um sistema que é microscopia *on-chip* holográfica sem lentes, que grava a sombra ou um "holograma" do objeto, que mais tarde pode ser reconstruída para observação. Dentro do sistema, é usado a técnica de super-resolução por pixel (além de outras técnicas computacionais) para conseguir criar uma imagem de alta resolução a partir de um conjunto de imagens de baixa resolução. No artigo, existem imagens que mostram o equipamento desenvolvido, fluxogramas mostrando cada passo para atingir a super-resolução e figuras que comparam e comprovam que ocorreu uma melhora na resolução. Como o artigo aborda e explica a técnica de super-resolução usada, é interessante para a pesquisa, mesmo que o foco não seja as células de sangue, mas sim exames de Papanicolau.

Os artigos de Mudanyali, Bishara e Ozcan (2011) e Greenbaum *et al.* (2013) demonstraram a aplicabilidade da técnica holográfica para melhorar a qualidade das imagens (super-resolução) de RBCs e de células de exames de Papanicolau, respectivamente. Como destaque, ambos os artigos propuseram equipamentos que não necessitam de lentes, o que reduz os custos de produção.

York *et al.* (2013) explicaram a diferença entre Esgotamento de Emissão Estimulada (STED) e Microscopia de iluminação estruturada (SIM), afirmando que a técnica SIM obtém resultados melhores com mais rapidez, sendo que ocorreu a implementação analógica da SIM, que dobra a resolução espacial de um microscópio fluorescente, sem comprometer velocidade, foto toxicidade ou campo de visão. A combinação de lentes dos equipamentos usados conseguiu melhorar a resolução, porém não utiliza algoritmos nem códigos, sendo focado para o hardware. E,

finalmente, o artigo focou em imagens 3D (rápidas e não invasivas) e hemodinâmica, não se concentrando para a microscopia das células de sangue.

Becatti *et al.* (2020) focaram na parte biológica, destacando em usar a super-resolução para encontrar formas parecidas entre células anômalas. A técnica usada, STED, mostrou modificações no diâmetro da fibra e na porosidade do coágulo. Porém, nem códigos nem algoritmos não são abordados, somente pouco citados. O artigo só afirma que usou super-resolução, além das células não serem de sangue.

Cai *et al.* (2020) explicaram a técnica de segmentação para reconhecer padrões de esferas, sendo chamada *Wavelet-based Segmentation for Spherical Images* (WSSA), sendo uma segmentação baseada em onduleta (*wavelet*) para imagens esféricas. Os autores relataram que para que geralmente ocorra o sucesso no *deep learning* precisa que aconteça a exploração das características de dados pela rede para que o aprendizado seja eficiente, como também a qualidade dos conjuntos de treinamento. No artigo, é enfatizado que o método proposto aborda a tarefa de segmentação com uma forma completamente diferente, não requerendo quaisquer conjuntos de treinamento. Os autores destacam, ainda, as principais contribuições: uma segmentação estruturada para imagens esféricas; uma estratégia iterativa com flexibilidade para adaptar o procedimento iterativo de acordo com os tipos de dados e recursos; além de onduletas esféricas, incluindo onduletas axissimétricos, onduletas direcionais e o híbrido direcional de onduletas curvilíneas. No artigo, existem as equações usadas, os resultados em gráficos e imagens esféricas como do globo terrestre que foram utilizadas como *data teste*, para depois serem testados em globo oculares para fazer uma recriação com os micros vasos observados na retina, além de apresentar um pseudocódigo mostrando qual o passo a passo desenvolvido pelo algoritmo (WSSA). Mesmo com todos os detalhes, o artigo e o algoritmo apresentados são focados em formas esféricas e não em células sanguíneas.

Zalevsky *et al.* (2011) mostraram o artigo dividido em duas partes, sendo somente a primeira parte interessante, pois utiliza a técnica de *geometric super resolution* que utiliza matrizes e máscaras binárias para alcançar a super-resolução. A técnica faz uma adição de uma máscara aleatória binária juntamente a imagem de baixa resolução. Os autores revelam que as vantagens de uma máscara é a redução da dependência de tolerâncias de fabricação. O objetivo da máscara, é superar a redução da resolução geométrica resultante da amostragem não ideal dos pixels, sendo ajustada adicionando equações, criando uma reconstrução “super-resolvida”

após a inversão de matriz adequada ao procedimento. O artigo contém imagens e equações que atingiram o resultado, porém não é utilizado para células de sangue.

Os últimos quatro artigos trataram sobre diferentes métodos e com objetivos distintos. Foram abordados desde o desenvolvimento de *softwares* especializados em análise e detecção de estruturas esféricas (CAI *et al.*, 2020) ou células anômalas (BECATTI *et al.*, 2020) até identificação e classificação de imagens em geral (ZALEVSKY *et al.*, 2011) e em 3D (YORK *et al.*, 2013). No entanto, focaram em melhorar a resolução de imagens em geral, mas não de células de sangue.

Em resumo, todos os oito artigos utilizaram técnicas que envolveram super-resolução. Por um lado, a maioria dos algoritmos de super-resolução foi empregada para imagens que não se concentravam em células sanguíneas. Por outro lado, alguns dos algoritmos não foram explicados detalhadamente, porém todos apresentaram resultados satisfatórios na qualidade para cada imagem requerida.

2.1.2 Técnicas para Imagens Vasculares

Todos os artigos nesta seção abordaram técnicas para observar hemodinâmica ou microvasos, sendo voltados para a parte vascular do sangue e não das células. A maioria das técnicas se encaixou no tema de ultrassom (ACKERMANN; SCHMITZ, 2015; DENCKS *et al.*, 2018; PIEPENBROCK *et al.*, 2019) e não observaram a lâmina de sangue. Alguns artigos destacaram mais a biologia (LAVINA; GAENGEL, 2015), como a reconstrução de vasos sanguíneos (CAI *et al.*, 2011; KOHLER *et al.*, 2014) e outros focaram na física, observando a velocidade e o volume do sangue que passa pelos vasos sanguíneos (YOUSEFI; QIN; WANG, 2013; OPACIC *et al.*, 2018; KIM *et al.*, 2019b).

Lavina e Gaengel (2015) descreveram diversas técnicas para observar a parte vascular de humanos e animais. Porém, o artigo não foca em nenhum algoritmo em específico e somente na parte biológica vascular em conjunto com as técnicas, não sendo a microscopia de células de sangue.

Cai *et al.* (2011) introduziram a técnica baseado em "*framelet*" de segmentação para obter o máximo de detalhes possíveis em tempo real, de forma automática e eficiente de estruturas tubulares. O artigo demonstra as fórmulas, as imagens e com pseudocódigo que foi usado para alcançar os resultados desejados.

Com resultados numéricos que mostram a capacidade do método para segmentação de estruturas tubulares, além de ser implementado rapidamente e obter limites ou superfícies muito precisas e suaves. No artigo, mesmo com a excelente explicação da técnica usada, ocorre o destaque de estruturas tubulares como vasos sanguíneos, não focando em células de sangue que é o foco desta pesquisa.

Piepenbrock *et al.* (2019) utilizaram uma técnica de microscopia de localização por ultrassom (ULM) com microbolha (MB), que depende da localização do subcomprimento de onda de MB individuais para reconstruir a micro vasculatura, sendo necessário que um vaso seja conduzido através de pelo menos uma MB. O artigo abordou a técnica de Imagem hemodinâmica de ultrassom de super-resolução baseada em escassez (SUSHI), que é um método usado na microscopia de localização foto ativada por fluorescência (fPALM) em combinação com Ultrassom com contraste aprimorado (CEUS) para obter imagens de alta resolução da vasculatura. Essa técnica foi combinada com algoritmos (Gauss, funções de espalhamento pontual -PSFs- dependentes da profundidade) para observar a sua precisão, concluindo que as abordagens baseadas em *deep learning* devem ser preferidas para obter uma eficiente capacidade computacional. Sendo assim, os métodos que são aplicáveis com altas concentrações de MB e os PSFs simulados dependentes da profundidade devem ser preferidos ao kernel Gauss ideal. Porém, mesmo que o artigo tenha discorrido sobre o algoritmo e técnica utilizada, tal técnica é empregada no ultrassom e não em microscopia de células de sangue.

Yousefi, Qin e Wang (2013) utilizaram Microangiografia Ótica (OMAG) baseado em Tomografia de coerência óptica (OCT) para fazer uma estimativa espectral de super-resolução para quantificar a hemodinâmica microvascular. O artigo combinou OMAG e OCT com outras técnicas para obter melhores resultados e maior velocidade, observando microvasos e a sua hemodinâmica em tempo real. O artigo também apresenta equações utilizadas para alcançar os resultados mostrados nas imagens (preto/branco e colorida), que demonstra a dinâmica do sangue de orelhas de ratos, indicando também a temperatura em que se encontra e como ela afeta a velocidade dos microvasos. Apesar do detalhamento do artigo, a técnica descrita é utilizada em tomografias e na hemodinâmica e não em microscopia de células sanguíneas.

Ackermann e Schmitz (2015) discutiram uma técnica que utiliza as posições de MB individuais, sendo detectados em super-resolução e o fluxo sanguíneo é

produzido estimando as velocidades das MBs a partir dessas posições. Nessa técnica, as MBs são usadas para o ultrassom em microvasos para alcançar a super-resolução e descobrir a velocidade do fluxo, sendo também para fazer a reconstrução de vasos (imagens). No artigo, são mostradas as equações utilizadas para detectar as MBs com tabelas e gráficos mostrando os resultados obtidos. Porém, o artigo utiliza a técnica para microvasos e hemodinâmica e não em células sanguíneas.

Dencks *et al.* (2018) explicaram a técnica de super-resolução de ULM que utiliza MB, tendo o objetivo de estimar o volume de sangue relativo (rBV) e conseguir fazer a reconstrução microvascular mais precisa e em menos tempo comparado com outras técnicas, além de conseguir informações como velocidade e direção do sangue. O artigo apresenta as equações e as imagens com os resultados e análises. Porém, não foca em microscopia de células de sangue e sim em imagens de ultrassom.

Kim *et al.* (2019b) mostraram a técnica de Microscopia fotoacústica de localização (L-PAM), a qual é baseada em "ondas ultrassônicas que são geradas pela expansão térmica transitória das moléculas absorvendo luz óptica". Sendo assim, utiliza ondas para recriar as imagens de sangue de animais e de humanos em vivo, podendo observar os microvasos e a hemodinâmica, sendo possível também localizar as posições dos contrastes endógenos para reconstruir as imagens localizadas. Porém, o artigo não possui nada em relação a códigos ou algoritmos usados.

Opacic *et al.* (2018) apresentaram uma técnica de super-resolução alternativa diferente de CEUS que é modelo de microscopia de localização por ultrassom (mULM) que, de acordo com o artigo, é uma técnica de rastreamento avançada adaptada para configurações clínicas. Com essa técnica, imagens de super-resolução e parâmetros novos podem ser retirados de forma rápida e prática, que facilita uma distinção precisa de tumores com diferentes fenótipos vasculares. Além de mostrar as diferenças na textura vascular de diferentes tumores, também determina a velocidade de fluxo dos vasos que correspondem a artérias ou veias. A técnica utiliza MB para ULM, sendo uma técnica da área de ultrassom, não focando em microscopia e nem o algoritmo é destaque, somente é citado que foi usado super-resolução e a atenção é em vasos sanguíneos e não em células de sangue.

Kohler *et al.* (2014) explicaram a técnica de *multi-frame* de super-resolução (SR) para observar o fundo da retina. A técnica reconstrói uma imagem de alta resolução com SNR aprimorado de vídeos com *multi-frame* de baixa resolução. Os

autores relatam que o trabalho possui três principais contribuições: ter inserido uma correção de iluminação para registro fotométrico para compensar a iluminação espacialmente e temporalmente heterogênea; ocorreu a avaliação de qualidade sem referência com o objetivo de fornecer pontuações de qualidade de imagem objetivas e selecionar parâmetros de reconstrução automaticamente; a verificação experimental que mostra a importância para diagnósticos através de câmeras de vídeo de fundo de olho. Com seções detalhadas mostrando as equações, fluxogramas com o processo, tabelas comprovando a melhora por meio de porcentagens e imagens que provam a melhoria na resolução no fundo da retina, mostrando um aumento na qualidade dos microvasos vistos. Porém, o artigo não apresenta o destaque em células sanguíneas.

Em suma, a maioria explicou de forma detalhada os algoritmos empregados, alguns focando mais nas texturas das imagens, outros destacando as velocidades e direções dos fluxos de sangue. Contudo, todos os nove artigos focaram em melhorar a resolução das imagens das estruturas dos microvasos sanguíneos e com imagens que comprovam o aprimoramento, sendo um campo de suma importância na área médica, mesmo não focando em células de sangue.

2.1.3 Técnicas para Imagens Celulares

Nesta seção foram abordados três artigos que destacam técnicas de super-resolução em células, não necessariamente sendo sanguíneas. Westmoreland *et al.* (2016) focaram em observar plaquetas e se possuem algum formato anormal. Pennanen *et al.* (2017) realçaram sobre como são e como identificar as estruturas dos osteoclastos e Jiang *et al.* (2020b) explicaram como usar a alta frequência para melhorar a imagem de uma amostra de sangue.

Westmoreland *et al.* (2016) explicaram que utilizam a Microscopia de super-resolução (SRM) para conseguir um diagnostico sem viés, confiável e fácil sobre o formato estrutural das plaquetas. A técnica utilizada (SIM) tem como objetivo identificar plaquetas de indivíduos saudáveis de plaquetas anormais. No artigo, existem imagens e gráficos demonstrando quanto a resolução foi melhorada. Porém, o artigo foca nos resultados e não em como foram feitos os algoritmos para alcançar tal resultado. Além disso, o resultado foi focado somente em plaquetas.

Pennanen *et al.* (2017) apresentaram a técnica STED para analisar osteoclastos (de humanos) que foram cultivados *in vitro*. A técnica apresenta imagens de resolução melhorada de curvas e ramificações dos osteoclastos em comparação com microscópica. Os autores afirmam que STED permitiu descobrir novas características da diversidade da arquitetura e da dinâmica em osteoclastos. No artigo, os resultados mostram a maior precisão proporcionada por uma microscopia STED, as imagens demonstram a melhor resolução, porém o foco não são células de sangue e sim os osteoclastos e o algoritmo de super-resolução somente é citado, não ganhando grande destaque.

Jiang *et al.* (2020b) explicaram a técnica da microscopia de extensão diagonal no domínio da frequência (FDDE), que usa o componente de alta frequência na direção diagonal para melhorar a resolução. Em microscopia de FDDE, a imagem é normalmente gravada com um sensor 2D com pixels em forma de grade. No artigo, ocorre a explicação das equações, imagens que demonstram os resultados alcançados e gráficos que mostram a quantidade dos pixels das imagens. O artigo foca na teoria, no algoritmo usado e o resultado atingido pelo mesmo, além disso, mostra a melhora de resolução em uma amostra de sangue (porém, não são as células).

Em síntese, a maioria dos artigos não detalhou os algoritmos usados, somente citando-os. Entretanto, em todos a teoria foi bem explicada, cada um com o uso de técnicas diferentes e com resultados que refletiram a melhora obtida.

2.1.4 Técnicas para Imagens de Células Sanguíneas

Os artigos apresentados nesta seção são os que abordaram as técnicas de super-resolução em microscopia de células de sangue. Alguns artigos apresentaram técnicas não adequadas de serem usadas por imagens adquiridas de banco de dados (LUO *et al.*, 2016; BHANDARI *et al.*, 2017; KIM *et al.*, 2019a), pois utilizaram um modo diferente de aquisição ou não observaram a lâmina microscópica para obter as células de sangue. Outros trataram de mostrar além do algoritmo como o desenvolvimento do *hardware* (ZHENG *et al.*, 2011; HUANG *et al.*, 2016a; LIU *et al.*, 2017; LI *et al.*, 2021b), pois, por serem CMOS ou *on-chip*, elas observam as “sombras” das células que passam, sendo assim, não possuem as imagens iguais às das lâminas de sangue

feitas através de um microscópio. Já os próximos três artigos (MAISELI *et al.*, 2014; HUANG *et al.*, 2016b; AYAS; EKINCI, 2020) realizaram um destaque especial sobre o *software*, podendo alguns terem potencial para serem implementados em *hardware*. E os últimos dois artigos da Pesquisa 01 (MA *et al.*, 2018; TOM *et al.*, 2019) são os mais similares ao que foi proposto para este trabalho de pesquisa, pois além de abordar o destaque para algoritmos de super-resolução com identificação e classificação de células sanguíneas, ocorre também a comparação entre arquiteturas de CNNs diferentes e de exemplos reais para observar a verdadeira precisão dos códigos propostos.

Kim *et al.* (2019a) apresentaram uma técnica de Imagem fotoacústica (PAI), sendo que ondas ultrassônicas foram induzidas pela expansão térmica transitória de moléculas que absorveram luz. Os autores relatam que o método de localização não apresenta restrição pelas partes de *hardware*, porém acaba requisitando agentes de contraste exógenos, como microesferas ou microbolhas, que são indesejáveis nas configurações de clínica. O foco foi uma nova Microscopia Fotoacústica (PAM) de localização livre de agentes de contraste e com sistema de um *scanner* galvanômetro (L-PAM-GS) para, através do processo de localização, melhorar a resolução temporal e, depois, a resolução espacial. No artigo, existem imagens do processo e dos resultados, sendo analisados microvasos de animais pequenos e de humanos *in vivo* e da hemodinâmica da orelha de um rato, e não nas células de sangue e por último, o algoritmo de super-resolução não recebe destaque.

Luo *et al.* (2016) explicaram de forma detalhada como uma super-resolução de pixel baseada na técnica de varredura de comprimento de onda, a qual gera reconstruções de alta resolução de baixas medições brutas amostradas e capturadas em vários comprimentos de onda dentro de uma faixa espectral estreita. Os autores afirmam que a fundamentação física dessa técnica é a alta correlação do comprimento de onda com os padrões de interferência subamostrados que existem em sistemas de imagem de difração, os quais são coerentes ou parcialmente coerentes, como, por exemplo, microscopia holográfica sem lente ou desfocada e também em sistemas de imagem baseados em lentes. A técnica apresenta velocidades rápidas de aquisição e precisa de menos medições para alcançar uma alta resolução, podendo até reconstruir imagens coloridas sem modificações no algoritmo. Assim, pode reduzir o custo de armazenamento e transmissão de dados, beneficiando a telemedicina e reconstruções remotas baseadas em servidor. O artigo mostra as equações utilizadas,

porém o algoritmo não é mostrado em grandes detalhes, pois é feito *on-chip* e é focado nas imagens dos resultados gerais do *hardware*, pois o *software* já está embarcado.

Bhandari *et al.* (2017) utilizaram a técnica de super-resolução que usa as ondas fotoacústicas (*Photoacoustic* - PA) para identificar e classificar as células sanguíneas. A técnica utiliza a característica espectral no domínio de Fourier, utilizando as frequências para a distinção, sendo que no efeito ondas fotoacústicas ocorre a conversão de energia eletromagnética para energia acústica, que manda informações como as dimensões e as densidades (dependendo de como foi propagado). O artigo explica em detalhes as funções e equações utilizadas, mostrando com gráficos os resultados obtidos. Porém, essa técnica possui o modo de aquisição diferente, não empregando o microscópio, sendo uma técnica inviabilizada de ser utilizada para o presente estudo.

Huang *et al.* (2016a) usaram a Rede Neural Convolutacional (CNN) baseada em *single-frame* super-resolução que através de imagens de baixa resolução da sombra da célula consegue alcançar imagens de alta resolução e, assim, identificar, classificar e contar as células sanguíneas. As CNNs têm sido usadas em *deep learning* para cuidar de grandes bases de imagens. O método de Super-Resolução baseada em Rede Neural Convolutacional (CNNSR) é rápido e computacionalmente eficiente com pouco pré-processamento ou otimização pós-processamento, tendo otimização no uso. No artigo, existem imagens que ilustram como é o equipamento, processo e fluxo, sendo um semicondutor de óxido metálico complementar (CMOS) sem a necessidade de lentes para observar as células. Por ser feito em *hardware*, o *software* precisa ser leve e eficiente. O desenvolvimento do algoritmo é bem detalhado, mostrando imagens que afirmam o melhoramento da resolução de células sanguíneas.

Liu *et al.* (2017) explicaram a técnica *single-frame* com CNNSR, a qual no método utiliza a sombra das células para serem processadas e detectar, classificar e contar. O algoritmo Seguidor de Origem de Rede Neural Convolutacional (SF-CNN) foi usado para melhorar ainda mais a resolução como, principalmente, de núcleos e das bordas das membranas que transformam imagens de baixa resolução em alta resolução, feito em *real-time*, de forma eficiente e com o mínimo de processador no *hardware* possível. O artigo apresenta desenhos, esquemáticos e imagens com os resultados obtidos, como o foco é o desenvolvimento em detalhes de um CMOS, o qual pode classificar e contar diferentes células sanguíneas, sendo o algoritmo usado

(CNN) um coadjuvante, não possuindo muito destaque, porém é explicado de forma eficiente e prática.

Li *et al.* (2021b) descreveram o desenvolvimento de um algoritmo baseado em super-resolução que identifica o formato "oval" das células de sangue, que precisa passar dos obstáculos como o efeito de difração e a baixa resolução, pois é um sistema *on-chip* de sensor de imagem CMOS (CIS) que não possui lentes. Assim, o processo compara os padrões de difração e de eliptócito para fazer de forma automática sem precisar de profissionais qualificados para identificar as células de sangue. No artigo, o algoritmo é dividido e explicado em quatro partes (segmentação da imagem, pré-processamento da imagem, super-resolução e base de julgamento para eliptócitos), sendo todas detalhadas com equações, imagens, fluxogramas para demonstrar a solução.

Zheng *et al.* (2011) relataram a criação de um *hardware* sem lentes e *on-chip* com Microscópio optofluídico de resolução de subpixel (SROFM) baseado em um algoritmo de super-resolução. No artigo, é explicado o processo do desenvolvimento tanto do *hardware* quanto do *software*, com esquemáticos, imagens e gráficos que relatam que foi possível evitar o efeito "*aliasing*" e melhorar a resolução da qualidade da imagem. Com o sistema desenvolvido, foi possível identificar as células de sangue normais e as infectadas pela doença da malária. E os autores enfatizam que, com tal equipamento, há a chance de o microscópio portátil ser feito em massa para ser comercializado, com o objetivo de detecção de doenças que envolvem deformação de células de sangue e parasitas transmitidos pela água.

Ayas e Ekinci (2020) explicaram a técnica de super-resolução de imagem única (SISR) que usa multi scale dCNNs para imagens de baixa resolução de microscópio que alcançam imagens com alta resolução. A SISR reconstrói uma imagem de alta resolução através da imagem de baixa resolução inserindo todas as altas frequências que faltam. Já as CNNs, são feitas de três camadas, que processam os dados e treinam para obterem maior precisão em encontrar as imagens corretas e que se conectam para obter a maior resolução. No artigo, existem imagens e tabelas mostrando os resultados obtidos e com quais parâmetros, mostrando o passo a passo de como funciona o algoritmo e sendo utilizado para melhorar imagens de amostras sanguíneas (mas sem classificar e nem identificar).

Huang *et al.* (2016b) esclareceram o desenvolvimento de um CMOS, de pouco custo e fácil manuseio, que utiliza algoritmos de super-resolução (ML baseado

em *single-frame* de super-resolução), sendo a Super-resolução baseada em máquina de aprendizado extremo (ELMSR) e CNNSR. Esses algoritmos são treinados com base de dados extensas, ambos sendo leves e o potencial para serem implementados em *chip*, com rede neural do modo "*feed-forward*" (computam de forma eficiente com pouca otimização pré ou pós-processamento). Visto que usam imagens de baixa resolução para conseguir reconstruir uma imagem de alta resolução, é possível identificar e classificar a célula de sangue e de tumor, através da sua própria sombra e sem a utilização de lentes. No artigo, ocorre o detalhamento de ambos os algoritmos de ML usados, enfatizando as vantagens e desvantagens de cada um, com pseudocódigos mostrando o processo, equações utilizadas e imagens para comparação, que acusaram que CNNSR teve menos ruído comparando com ELMSR, com as bordas que não estão borradas e com 9,5% de qualidade melhorada em relação ao outro.

Maiseli *et al.* (2014) desenvolveram um algoritmo que utiliza super-resolução, não especificando qual a técnica usada. O diagnóstico seria todo automatizado para que melhorasse as resoluções e depois identificasse as células normais e anormais para detecção de malária. No artigo, existem fluxogramas e processos que mostram o passo a passo para ocorrer o melhoramento da resolução, também contém a comparação entre detectar malária com uma imagem com pouca resolução e com alta resolução, a qual demonstravam através de imagens que o algoritmo realmente conseguiu identificar melhor as células infectadas com a melhor resolução e com gráficos e correlações que mostram o nível de precisão alcançado pelo código desenvolvido. Resumindo, o artigo aborda um algoritmo de super-resolução (detalhado) e identificação de um tipo de célula de sangue (RBCs).

Ma *et al.* (2018) explicaram de forma comparativa os tipos de algoritmos para *single-frame super-resolution reconstruction* baseado em CNNs, explicando as equações, tabelas, fluxogramas e imagens para o que precisa ser feito, citando que foi feito para *hardware* (CMOS). Porém, o *hardware* não é o foco do artigo. A técnica de Rede de Super-resolução de Células (CSRNet) resolve o problema de baixa resolução e poucas informações em imagens de células que foram coletadas por sistema de detecção de células sem lentes, criando melhores imagens de alta resolução e com detalhes (comparada a outras técnicas). O artigo é focado em comparar os algoritmos utilizados para as células sanguíneas (RBC) para observar qual atinge a melhor resolução, comprovando o resultado através de imagens.

Tom *et al.* (2019) utilizaram a técnica SISR com CNN para alcançar a super-resolução. Propõem o uso de uma rede de super-resolução (SRNet) treinada para minimizar a perda de reconstrução entre as imagens reais (não artificiais) e as imagens com SR e que usa as redes de teste visual relativístico de Turing (rVTT) para diferenciar entre pares de imagens reais e com SR, como também entre pares de manchas em imagens reais e imagens com SR que são selecionadas da região de interesse. O objetivo está em aumentar a porcentagem de diferenciação do algoritmo de identificação de núcleos e citoplasma e de ser capaz de restaurar a textura relevante dentro do núcleo e ao redor do citoplasma nas células. O artigo apresenta imagens, equações e tabelas que explicam de forma detalhada todo o processo feito dos algoritmos na microscopia de células de sangue.

Em resumo, todos os doze artigos utilizaram técnicas que envolveram super-resolução focadas em células sanguíneas. Alguns focaram mais em *hardware*, outros mais no *software* com as explicações dos processos com detalhes. Além de que alguns compararam os algoritmos e equipamentos com os de outros autores para observar a distinção e comprovar a melhora feita por eles. Todos tinham o mesmo objetivo de melhorar a resolução de imagens de células de sangue. E, alguns, até se sobressaem em identificar e classificar as células de forma precisa.

Assim, após toda a seção da Pesquisa 01, tem-se um destaque especial para o último tópico, mais especificamente para trabalhos mais semelhantes à proposta desta pesquisa que são dos autores Ma *et al.* (2018) e Tom *et al.* (2019) (ver Quadro 2). Além de abordar o destaque para algoritmos de super-resolução com identificação e classificação de células sanguíneas, ocorre também a comparação entre arquiteturas diferentes e de exemplos reais para observar a verdadeira precisão dos códigos propostos, conforme apresentado anteriormente nessa seção.

Quadro 2 - Destaques da Pesquisa 01

Ma et al. (2018)	Sangue	CSRNet & FSRCNN	X	X
Tom et al. (2019)	Sangue	SISR & CNN	X	X

Fonte: autoria própria.

2.2 PESQUISA 02

A Pesquisa 02 apresenta os artigos que foram filtrados de três bases diferentes com o objetivo de abordarem métodos (principalmente algoritmos) com o tema de melhoramento de resolução de imagens microscópicas ou que demonstrassem técnicas de identificação e classificação de células sanguíneas. Foram essenciais para criar a base fundamental para desenvolver o algoritmo desta pesquisa. Assim, foram divididos em seis tópicos. O primeiro é “Melhorar Imagens Médicas” com aproximadamente 2,5% dos artigos da Pesquisa 02; o segundo é “Imagens de Células em Geral” com $\cong 7,4\%$ dos artigos; o terceiro é “Imagens de RBCs” com $\cong 27,2\%$ da pesquisa; o quarto é “Imagens de WBCs” com $\cong 8,6\%$ das leituras; e o último é “Imagens de Todas as Principais Células Sanguíneas” com aproximadamente 11,1% do total, os quais são detalhados nos próximos tópicos

2.2.1 Técnicas para Melhorar Imagens Médicas

Conforme o Quadro 3, foram identificados dois trabalhos que abordaram o tema de melhorar imagens médicas ou biomédicas com o auxílio de computadores. Selzer (1968) abordou o início do uso de computadores para filtrar imagens biomédicas para observar detalhes e ajudar um diagnóstico mais preciso. Já Vidyasaraswathi e Hanumantharaju (2015) trataram a técnica de equalização de histograma (HE) para melhorar a qualidade das imagens, principalmente, as imagens ruidosas, auxiliando no estágio inicial do diagnóstico e que são detalhados a seguir.

Quadro 3 - Visão Geral dos Artigos da Pesquisa 02 da seção Técnicas para Melhorar Imagens Médicas

Artigo	Tema/Categoria	Técnicas propostas	Técnicas comparadas	D	A	C
Selzer (1968)	Melhorar imagens biomédicas com computadores	-	-	-	X	-
Vidyasaraswathi e Hanumantharaju (2015)	Melhorar imagens médicas	HE	-	-	X	-

*D = Detalhamento das técnicas (Algoritmos)

*A = Aplicações microscópicas para células sanguíneas

*C = Classificação/Identificação de células sanguíneas

Fonte: autoria própria.

Selzer (1968) abordou o tema de melhoramento de imagens através de computadores, descrevendo principalmente as técnicas de filtragem. O autor relata que as imagens com a ajuda de computadores demonstram menores perdas da imagem com melhoria na resolução e que se ocorrer o aprimoramento dos computadores, a imagem alcança um nível ainda melhor que o anterior. O autor revela, ainda, que quando a imagem é aprimorada, as características essenciais são mais facilmente vistas do que na imagem original. Sendo assim, são mais confiáveis, além de descrever que o uso de *scanners* controlados por computador e a obtenção da exibição das imagens *online* podem reduzir o tempo de processamento de forma substancial. Entretanto, o autor avisa que esses tipos de sistemas são muito caros. Mas, está confiante de que os *scanners* terão mais disponibilidade à medida que as aplicações como o melhoramento de imagens através de computadores aumentarem em conjunto.

Vidyasaraswathi e Hanumantharaju (2015) desenvolveram um algoritmo que melhora imagens usando técnicas de equalização de histograma (HE), sendo imagens que foram capturadas em iluminações não favoráveis, embaçadas ou ruidosas. O algoritmo faz uma implementação sucinta de diferentes técnicas que melhoram as imagens, como: HE, *Adaptive HE*, *Histogram Modified Contrast Limited Adaptive HE*, *Brightness Preserving Dynamic HE*, *Sharp Contrast Limited Adaptive HE*. O método escolhido foi um conjunto de *Iterative Dynamic*, *Dualistic Sub Image HE* e *Brightness Bi-HE*. Os autores ressaltam que o desenvolvimento desses algoritmos ajudaria os médicos a diagnosticar a doença no estágio inicial apenas, sendo para melhorar o desempenho de sistemas de processamento de imagem, mais compactos, de baixo consumo de energia e de alta velocidade para reduzir custos e tempo.

Os dois artigos relataram os métodos utilizados para melhorar a qualidade das imagens. Entretanto não se aprofundaram nos algoritmos usados. Porém, é demonstrado nos resultados que ocorreu a melhoria de imagens na área da saúde com o auxílio de computadores e seus algoritmos (SELZER, 1968; VIDYASARASWATHI; HANUMANTHARAJU, 2015).

2.2.2 Técnicas para Imagens de Células em Geral

No Quadro 4 estão apresentados os artigos desta seção que abordaram técnicas que detectam, identificam, segmentam ou classificam células, podendo ser ou não sanguíneas, sobrepostas ou aglomeradas, saudáveis ou anormais.

Quadro 4 - Visão Geral dos Artigos da Pesquisa 02 da seção Técnicas para Imagens de Células em Geral

Artigo	Tema/Categoria	Técnicas propostas	Técnicas comparadas	D	A	C
Lane <i>et al.</i> (2018)	Sangue - Antígenos (RBC e Plaquetas)	BloodTyper	-	X	X	-
Liu <i>et al.</i> (2020)	Sangue - células diferentes	TIMELY	KNN/ KNCN/ CleanLab	X	X	X
Jiang <i>et al.</i> (2020a)	Sangue - Células (doenças)	GFS-ExtremeNet	Faster R-CNN/Mask R-CNN/SSD/YOLOv3/CornerNet/GFS-CornerNet/ExtremeNet	X	-	X
Long <i>et al.</i> (2021)	Sangue - Células diferentes (Doença)	BloodCaps	VGG16/AlexNet/ResNet-18/InceptionV3	X	X	X
Song e Wang (2009)	Sangue - células sobrepostas	Morphological Techniques	-	X	X	X
Nilufar e Perkins (2014)	Sangue - células Geral	Standard Snake	-	X	X	-

*D = Detalhamento das técnicas (Algoritmos)

*A = Aplicações microscópicas para células sanguíneas

*C = Classificação/Identificação de células sanguíneas

Fonte: autoria própria.

O primeiro artigo focou em identificar antígenos no sangue (LANE *et al.*, 2018). O segundo, em reconhecer os rótulos em que as células foram marcadas e verificar se estão corretos (LIU *et al.*, 2020). O terceiro artigo focou em classificar qualquer tipo de célula e até avaliar se possui algum tipo de doença (JIANG *et al.*, 2020a). O quarto

focou em multi classificar diferentes tipos de células sanguíneas (LONG *et al.*, 2021). Já o quinto focou em distinguir células sanguíneas que estão sobrepostas, aglomeradas ou que se encostam (SONG; WANG, 2009). Por fim, o último focou na forma como detecta e reconstrói os contornos dos objetos desejados, podendo ser também células sanguíneas (NILUFAR; PERKINS, 2014).

Lane *et al.* (2018) desenvolveram um algoritmo chamado de BloodTyper baseado em sequenciamento do genoma completo para identificar antígenos de hemácias e plaquetas. O algoritmo utilizou uma combinação de sequência de gene, sequência de cobertura, análise do número de cópias e detecção para selecionar os alelos do antígeno corretos. Esses alelos foram integrados para determinar o fenótipo do antígeno, que o algoritmo final validou em 200 genomas do estudo. O Excel foi usado para calcular estatísticas de desempenho como sensibilidade, especificidade, valor preditivo positivo, valor preditivo negativo e precisão. O algoritmo alcançou uma alta precisão de identificação de antígenos com valores próximos de 100%.

Liu *et al.* (2020) elaboraram um algoritmo que consegue identificar se a classificação feita da célula está correta ou não, ajudando no desafio da inconsistência do rótulo, chamado de TIMELY. A primeira etapa do algoritmo é identificar os rótulos incorretos, comparando com as características morfológicas da célula e a segunda etapa é oferecer um rótulo que se adequa as tais características, comparando com outras células que são vizinhas. O método proposto usa uma rede neural que aprende as características principais, que com os algoritmos de inferência de pseudotempo, usa tais características ao longo de uma trajetória e estima os pseudotempos, que combina com árvores de Markov ocultas não homogêneas, na qual os rótulos correspondentes são classificados conforme as observações. Por fim, a imagem é demonstrada com os rótulos incorretos e com os propostos (sem mudar diretamente). Para a avaliação foi usada duas bases de dados de WBCs com rótulos ruidosos, comparando com outros três métodos, sendo k-vizinhos mais próximos (KNN), k-mais próximos dos centroides vizinhos (KNCN) e Cleanlab, e contrapondo rótulos propostos com os rótulos *ground-truth* para calcular a precisão. Com os resultados demonstrando que foi ultrapassado os outros métodos em termos de identificação precisa e correção de rótulos ruidosos.

Jiang *et al.* (2020a) desenvolveram um algoritmo baseado em *deep learning* com espectro de características geométricas em ExtremeNet, chamado de GFS-ExtremeNet, responsável por reconhecer células com o uso de características

geométricas. A forma e o tamanho da célula podem variar consideravelmente. Com isso, pelo algoritmo ser baseado no formato, resulta na detecção precisa de qualquer alvo celular. Os autores ressaltam que o método foi ainda capaz de detectar parasitas unicelulares dentro das RBCs sem diagnóstico incorreto do outro, além de parasitas como *Babesia*, *Toxoplasma* e *Trypanosoma*, mostrando a capacidade de detectar tipos múltiplos de células e destacaram que é feito tudo com alta precisão. ExtremeNet é considerado o melhor modelo em termos de performance e acurácia comparado com outras estruturas de *deep learning*, sendo colocada uma tabela com comparação de precisão entre elas. Os autores destacam que a precisão é altamente dependente da marcação do ponto extremo na área inata de dados de treinamento, Assim, precisa ser demarcado de forma correta para ter melhores resultados. Finalizando, os autores enfatizam que o método proposto é de extrema importância na criação de um sistema aprimorado de detecção automática de células.

Long *et al.* (2021) criaram um algoritmo para multiclassificar diferentes tipos de células sanguíneas sendo uma rede de arquitetura de cápsula modificada, chamada de BloodCaps, identificando oito tipos: basófilos, eosinófilos, linfócitos, neutrófilos, monócitos, granulócitos imaturos (incluindo metamielócitos, mielócitos e promielócitos), plaquetas e eritroblastos. O método proposto foi comparado com outras técnicas como: modelos VGG16, AlexNet, ResNet-18 e InceptionV3, além de serem usados três tipos de base de dados (imagens com baixa resolução, base com poucas imagens e ambos). Para avaliação, foi usado o método K-fold cross com as métricas de precisão, sensibilidade, especificidade, precisão e F1-Score para a parte quantitativa. Os resultados mostraram que o método proposto se sobressaiu comparado com todos os outros métodos, além de apresentar os melhores resultados também em todas as bases de dados utilizadas.

Song e Wang (2009) arquitetaram um algoritmo responsável por separar células sanguíneas que estão sobrepostas, aglomeradas ou que se encostam, sendo baseado na informação do formato das células. O algoritmo possui essas etapas: melhorar a imagem; encontrar o limiar da imagem; preencher internamente as células; obter a célula do sangue; separar os aglomerados de células sanguíneas; identificar os contornos dos aglomerados de células sanguíneas; fazer a separação detectando pontos interessantes na aglomeração e identificar as linhas de separação apropriadas. Outros métodos sobre separação de células sanguíneas por sobreposição são baseados na forma das células sanguíneas, começando por detectar os pontos

interessantes para separação, construir as linhas de separação possíveis e verificar as linhas de separação por restrições. Os autores afirmam que o algoritmo procura a célula sanguínea com uma forma essencialmente elíptica, não alongada e que cada célula tem o tamanho em uma certa faixa. O algoritmo, para ter mais precisão, precisa obter as linhas de separação apropriadas. Por isso, algumas restrições devem ser feitas, sendo elas: identificar os aglomerados por meio da aproximação do polígono, depois detectar os pontos interessantes sobre os contornos, para depois construir e verificar as linhas de separação. O algoritmo conseguiu alcançar os objetivos pedidos, porém os autores afirmam que precisa de mais métodos sofisticados para ter mais precisão em encontrar as linhas de separação entre as células sobrepostas.

Nilufar e Perkins (2014) estruturaram um algoritmo que detecta e constrói o contorno do objeto desejado, combinando com o padrão de classificação probabilística, podendo aprender a identificar automaticamente os limites dos objetos desejados em imagens microscópicas, como imagens de microscopia de luz de sangue células e de uma seção de tecido muscular, além de elétron microscopia. O algoritmo usa o método cobra (*snake*), identificando uma boa função de custo e evita o ajuste manual de parâmetros, com otimização eficiente e aprendizado de uma função de custo. Os autores revelam uma desvantagem que a malha de programação dinâmica não pode capturar objetos de fortes concavidades e sendo preciso empregar um sistema mais poderoso de programação dinâmica que superaria tais dificuldades.

Os seis artigos descreveram métodos usados para detectar e/ou classificar imagens das células (sendo sanguíneas ou não), porém a maioria não possui foco nas principais células sanguíneas (RBC, WBC e plaquetas) ou focaram em doenças específicas ou destacaram células sobrepostas ou sobre contornar os objetos desejados. Entretanto, alguns artigos explicam os algoritmos usados e quais os passos seguidos, sendo de grande importância para esta pesquisa.

2.2.3 Técnicas para Imagens de RBCs

De acordo com o Quadro 5, nesta seção são exibidos os artigos que fornecem técnicas para detectar e reconhecer as RBCs em imagens microscópicas. Os primeiros autores citados se destacaram por usar alguns outros *datasets*, além de testar somente com RBCs (HUANG *et al.*, 2014; SONG; XIAO; LIAN, 2018). Os seguintes autores utilizaram métodos para identificar doenças que afetam as hemácias, como a falciforme, eliptocitose, entre outros: Elsalamony (2016, 2017); Xu *et al.* (2017); Grochowski *et al.* (2019); Petrovic *et al.* (2020); Zhang *et al.* (2020); Lebel *et al.* (2021).

Já outros autores focaram em separar as RBCs aglomeradas ou sobrepostas, podendo ser ou não saudáveis. Entre eles, Kanafiah *et al.* (2015); Wei *et al.* (2015); Wei e Cao (2016); Abu-Qasmieh (2018). Alguns autores utilizaram os formatos que as hemácias apresentam para identificá-las, como Frejlichowski (2012) e Sarrafzadeh *et al.* (2020). Outros autores usaram diversas técnicas que são diferentes entre si. Como por exemplo, a técnica K-Mean Clustering, apresentada por Savkare, Narote e Narote (2016); a técnica de diferenciação de cores, apresentada por Abbas (2015); a técnica FDCT, apresentada por Shirazi *et al.* (2015); a técnica RCNN, apresentada por Qiu *et al.* (2020); e a técnica SVM apresentada por Hortinela *et al.* (2020). Embora sejam técnicas diferentes, todas possuem o mesmo objetivo de classificar as RBCs.

O penúltimo artigo somente usou técnicas para aumentar a quantidade de imagens no banco de dados para treinar melhor os algoritmos (BAILO *et al.*, 2019). Por fim, o último artigo demonstrou um sistema composto por *hardware* e *software* para captar as características das RBCs e classificá-las (LI *et al.*, 2021a).

Quadro 5 - Visão Geral dos Artigos da Pesquisa 02 da Sessão Técnicas para Imagens de RBCs

Artigo	Tema/Categoria	Técnicas propostas	Técnicas comparadas	D	A	C
Song, Xiao e Lian (2018)	Sangue - RBC e com outros datasets (Encontra as células, somente)	CSRCM	-	X	X	X
Huang et al. (2014)	Sangue - RBC e HepG2	ELMSR	Comercial	X	X	X
Zhang et al. (2020)	Sangue - RBC (doença)	dU-Net	Region Growing/ Ilastik/U-Net/PSPNet/DeeplabV3+	X	X	X
Elsalamony (2017)	Sangue - RBC (doença) (formas)	CHT	-	X	X	X
Lebel et al. (2021)	Sangue - RBC (doença)	Deep-UV Illumination + CNN Resnet-50	-	X	X	X
Xu et al. (2017)	Sangue - RBC (doença)	dCNN	-	X	X	X
Elsalamony (2016)	Sangue - RBC (doença)	CHT + Watershed + Morphological Techniques	-	X	X	X
Petrovic et al. (2020)	Sangue - RBC (doença)	Review	-	X	X	X
Grochowski et al. (2019)	Sangue - RBC (doença e saudável)	Eigenfaces + NN	-	X	X	X
Wei e Cao (2016)	Sangue - RBC (individual ou sobreposta)	K-means + Morphological Techniques	-	X	X	X
Kanafiah et al. (2015)	Sangue - RBC (sobreposta)	RCF	Watershed/ Granulometry	X	X	X
Wei et al. (2015)	Sangue - RBC (aglomeradas)	Watershed + K-means	-	X	X	X
Abu-Qasmieh (2018)	Sangue - RBC (aglomerada)(doença)	DLSF	IRHT	X	X	X
Sarrafzadeh et al. (2020)	Sangue - RBC (formato)	FCT	CHT-based	X	X	X
Frejlichowski (2012)	Sangue - RBC (formato)	Morphological Techniques	-	X	X	X
Savkare, Narote e Narote (2016)	Sangue - RBC	K-Mean clustering + Watershed	-	X	X	X
Abbas (2015)	Sangue - RBC	Morphological Techniques	-	X	X	X
Shirazi et al. (2015)	Sangue - RBC	FDCT + Gram-Schmidt + Snake	-	X	X	X

Quadro 5 - Visão Geral dos Artigos da Pesquisa 02 da Sessão Técnicas para Imagens de RBCs (Cont.)

Artigo	Tema/Categoria	Técnicas propostas	Técnicas comparadas	D	A	C
Qiu et al. (2020)	Sangue - RBC (aglomerada)(doença)(poder ser)	Faster - RCNN + ResNet + Gradient Boosting	Watershed/U-Net	X	X	X
Hortinela et al. (2020)	Sangue - RBC	Watershed + SVM	-	X	X	X
Bailo et al. (2019)	Sangue - RBC	GAN	-	X	X	X
Li et al. (2021a)	Sangue - RBC	Morphological Techniques	-	X	X	X

*D = Detalhamento das técnicas (Algoritmos)

*A = Aplicações microscópicas para células sanguíneas

*C = Classificação/Identificação de células sanguíneas

Fonte: autoria própria.

Song, Xiao e Lian (2018) desenvolveram um algoritmo responsável por segmentar célula/núcleo que usa a estrutura de aprendizagem de pares contornosamente que transforma as atividades de detecção e segmentação em termos de um problema de regressão unificada (local e limite). O algoritmo proposto é um modelo de cadeia de regressão esparsa em cascata (CSRCM), que calcula a localização e o reconhecimento de cada objeto, além da detecção de limite de agrupamento dos objetos. O modelo primeiro aprende um conjunto de características online, antes do treinamento em cascata. No CSRCM, o método atualiza de forma iterativa até a convergência a localização e o limite dos objetos agrupados, sendo que os testes mostram a relação conectada entre as características extraídas e pontuações estimadas que aumentam o desempenho de detecção e segmentação de células/núcleos. Os autores ressaltam que as principais contribuições do método proposto são a estrutura unificada, o modelo de CSRCM e os experimentos. Os autores apresentaram, também, a eficácia do método proposto comparando com outros métodos e com outros conjuntos de dados de microscopia.

Huang *et al.* (2014) desenvolveram um citômetro microfluídico baseado em imagem de contato com o *software* de ELMSR com *single-frame* para reconhecer e contar células em uma solução de fluxo contínuo, sendo amostras de RBC e HepG2. O equipamento usa LED que ilumina por cima as células que passam, para o CMOS captar a imagem da sombra embaixo, sendo depois rapidamente processadas por ML baseado em *single-frame* SR, que pode melhorar resolução de imagens de sombra

de modo que se possa reconhecer e conte as células que fluem. O algoritmo possui três etapas, sendo a primeira a detecção de célula de fluxo baseada em diferenciação temporal; a segunda parte é o reconhecimento do tipo de célula baseado em ELMSR de *single-frame* (que melhora a qualidade da resolução da imagem); e a terceira é a contagem de células de cada tipo. Os autores ressaltam que o fator de aumento de resolução maior pode ser selecionado, mas precisa de maior tempo de processamento e complexidade. E para validação, foi comparado com outro sistema comercial de fluxo de citômetro, mostrando ter menos de 8% de erro.

Zhang *et al.* (2020) elaboraram um algoritmo baseado em *deep learning* de segmentação semântica para classificação de RBCs de pacientes que possuem doença falciforme, o algoritmo é feito por camadas deformáveis de convolução para a estrutura U-Net clássica e implementa o U-Net deformável (dU-Net). O algoritmo usa segmentação binária e segmentação semântica de multiclases e foi avaliado a performance do dU-Net no nível de pixel e de células, comparando o método proposto com métodos de segmentação não-supervisionados (*region growing*, *Ilastik*) e supervisionados (U-Net, PSPNet, DeeplabV3+). O algoritmo classificou em quatro classes com forma e textura parecidos: discócitos e oval; alongado e foice; granular; e por última, reticulócitos, estomatócito, equinócitos. Os autores afirmam que os resultados experimentais demonstraram que a performance ultrapassou outros métodos de segmentação, sendo mais robusto para as variações no tamanho da célula, textura e forma, sendo refletido na capacidade de segmentar a borda da célula, separar células em contato, discriminar objetos de ruído de células reais e garantir a integridade das formas celulares.

Elsalamony (2017) criou um algoritmo capaz de identificar as formas de RBCs normais e as que causam anemia (falciforme e eliptocitose) baseado na Transformada de Hough Circular (CHT) e nas formas geométricas que são as "assinaturas" de cada tipo de RBC. O CHT é responsável por reconhecer as RBCs e se ocorrer de aparecer células sobrepostas, usar o algoritmo Watershed para separar e a função morfológica para classificar, além de ter o processo de retirar os ruídos das imagens. O artigo apresenta o passo a passo do algoritmo proposto com detalhes e como foi feito o processo de cada etapa. E para avaliar o desempenho do algoritmo proposto foi usada a Rede neural (NN), sendo que a classificação depende das variáveis como: as células de entrada, perímetro, excentricidade, área convexa, área, proporção, solidez; e com três métricas que são: acurácia na classificação, especificidade e sensibilidade. O

autor afirma que a eficácia do algoritmo proposto pode ter aumentado a melhoria do diagnóstico de anemia de acordo com o alcance de 100% no processo de detecção em relação às amostras colhidas.

Lebel *et al.* (2021) elaboraram uma técnica que faz a classificação de RBC com parasita responsável pela malária sem conter fixação ou coloração e com o objetivo de alcançar a alta performance. O objetivo é demonstrar que é possível a implementação de imagem livre de rótulos e classificação de *P. falciparum* vivo (malária) que permitirá maior reduções no tempo do técnico, treinamento, custo do reagente e variabilidade do observador, sem perder a precisão e a performance. O artigo faz o uso de dois microscópios. O primeiro possui iluminação visível e quase-ultravioleta e o segundo, que foi feito sob encomenda, possui iluminação ultravioleta profunda a fim de explorar o desempenho em uma variedade de condições de imagem. A parte do algoritmo, mais especificamente a segmentação semântica, foi realizada por uma CNN ResNet-50 sendo treinado com RBCs com diversas aparências e estágios do ciclo de vida do parasita. A validação foi feita por análise estatística de matrizes de confusão. O artigo apresenta seções separadas (ao final) para explicar o *hardware* e o *software* empregados com mais detalhes.

Xu *et al.* (2017) desenvolveram um algoritmo baseado em deepCNNs (dCNN) para detectar e classificar automaticamente RBCs que apresentam a doença falciforme para auxiliar na severidade da doença do paciente e ter intervenções estratégicas por tratamento personalizado. O algoritmo RBC-dCNN possui quatro etapas principais de treinamento: extração hierárquica das divisões das RBCs; normalização das divisões da RBCs invariante do tamanho; classificação de padrão RBC e cálculo automatizado do fator de forma RBC. O primeiro passo é responsável por encontrar e separar as RBCs em divisões, com o resultado de "pedaços de imagens" com RBCs individuais. A segunda etapa faz a seleção inicial da máscara de normalização, depois realiza a geração da máscara de normalização adaptativa com base no mapeamento linear de intensidade estatística para finalmente alinhar essa máscara com a divisão de imagem com RBC correspondente. O terceiro passo faz a classificação das RBCs seguindo para a quarta etapa que faz a análise automatizada do fator de forma RBC após classificação do formato. O artigo apresenta fluxogramas, imagens, equações para demonstrar o desenvolvimento do sistema. No final, as divisões de imagens das RBCs foram classificadas em oito categorias com a utilização da validação cruzada K-fold para ser uma forma eficaz de medir o desempenho

preditivo para o modelo dCNN e utilizaram diferentes métricas para fazer uma avaliação final da performance sendo colocadas em tabelas e gráficos para melhor comparação, mostrando ter um ótimo desempenho.

Elsalamony (2016) criou um algoritmo para detectar RBCs saudáveis e não-saudáveis (eliptocitose, células falciformes, microsites e formato desconhecido) usando CHT, que pode detectar até se as células estiverem sobrepostas ou nas bordas das imagens ou escondidas, Watershed e parte de morfologia (ambos usados para melhorar e separar células sobrepostas durante o processo de segmentação), para identificar o formato das células, usando área, área convexa, perímetro, excentricidade, solidez e proporção como parâmetros. A NN foi aplicada para testar e analisar o desempenho do algoritmo proposto para diagnosticar e determinar se um paciente tem um ou mais tipos de anemia, sendo a classificação dependente das variáveis de dados das células detectadas. No artigo, os resultados experimentais são apresentados em dois modos: a primeira forma é a detecção de RBCs saudáveis e não saudáveis e a contagem dessas células; e a segunda forma é a análise por NN da classificação dos dados das células. Por fim, o autor ressalta que os resultados experimentais demonstraram alta precisão e taxas de sucesso na análise de células saudáveis e não saudáveis com o desempenho calculado por três medidas estatísticas: precisão de classificação, sensibilidade e especificidade.

Petrovic *et al.* (2020) elaboraram uma metodologia para selecionar a classificação de métodos e características com melhor desempenho para suporte de diagnóstico através de imagens de lâminas de sangue focando em RBCs, principalmente para o diagnóstico de doença falciforme. Os autores coletaram e implementaram as características e classificadores que a literatura existente mostrou, comparando com o caso de estudo ou com casos semelhantes. As etapas se constituíram de pré-processamento, segmentação, extração da característica e classificação, sendo cada passo explicado com fluxogramas para ilustrar o funcionamento e com tabelas para descrever cada característica extraída, cada tipo de classificador com suas vantagens e desvantagens, explicando como selecionar os melhores parâmetros para cada classificador. Em seguida, foi mostrado de que maneira selecionar as características mais importantes para classificação, a fim de diminuir a complexidade e tempo de treinamento. Por último, os autores ressaltam que o método proposto apresentou melhores resultados comparados com semelhantes da literatura.

Grochowski *et al.* (2019) estruturaram um algoritmo capaz de identificar e contar as hemácias saudáveis e patológicas de sangue humano e canino. É necessário remover os ruídos do fundo da imagem e destacar os centros das células saudáveis. Assim, são usadas operações morfológicas como dilatação e erosão, pois permitem a separação visual de células sobrepostas. Já para extrair as características das células de sangue, é utilizado o algoritmo *Eigen faces*, sendo que, os autores ressaltam que a diferença entre as células saudáveis e patológicas são somente as formas das bordas das células e é mostrado o passo a passo de como é a reconstrução da RBC usando esse algoritmo. Para classificação, a NN foi usada com o passo a passo detalhado e com fluxograma do sistema por completo. Com o objetivo de aumentar a confiança dos resultados, o método de validação cruzada K-fold foi aplicado. O artigo apresenta tabelas com os resultados da contagem e da classificação baseados na acurácia, sensibilidade e especificidade tanto do sangue humano quanto canino. Os autores afirmam que o equipamento de medição óptica aplicado em conjunto com técnicas avançadas no processamento de imagens, estatística multivariada (*Eigen faces*) e algoritmos inteligentes (redes neurais) alcançaram resultados bons tanto na parte de separar, contar e por fim, classificar as RBCs. Os autores afirmam que as vantagens desses métodos, em contrapartida com um operador humano, são a necessidade de menos tempo para examinar grandes quantidades de dados, bem como também sua repetibilidade e objetividade. Entretanto, esses métodos são dependentes de dados, logo, sua eficácia depende da qualidade e da quantidade dos dados disponíveis para treinamento. Por isso, o formato e a validação necessitam ser supervisionado e sustentado por médicos especialistas.

Wei e Cao (2016) arquitetaram um algoritmo automático para segmentação de RBCs que estão complexamente sobrepostas. No método proposto, as RBCs são extraídas e o formato bicôncavo é usado para prever a posição do RBC em clusters, e também é utilizado o algoritmo K-means para detectar a borda interna de sobreposição das RBCs. O algoritmo é composto por algumas etapas sendo pré-processamento (usa componentes de cor para fazer binarização, ler os pixels e guardar sua localização), classificação (divide em RBCs individuais ou sobrepostas), segmentação (faz aglomeração dos pixels, detecção das bordas e restauração, previsão e, enfim, a segmentação) e contagem (separação das células sobrepostas, usa K-means para ajuntamento, detecção das bordas, separação das células internas

e, finalmente, contagem). A avaliação foi feita pelas métricas de precisão e acurácia, além de ser comparado com outros métodos, demonstrou ter um desempenho e performances melhores.

Kanafiah *et al.* (2015) elaboraram um algoritmo responsável pelo processo de separação e distinguir células sobrepostas (RBCs e exames de Papanicolau), conhecido como formação de células com base radial (RCF). O processo do algoritmo é feito por algumas etapas: segmentar a imagem em duas regiões (as células sobrepostas e o fundo), mudar para uma imagem binária, determinar os pontos de segmentação com as coordenadas, determinar todos os pixels localizados nas bordas das células sobrepostas, calcular o raio dos pixels que estão nas bordas para encontrar cada célula sobreposta. O algoritmo também possui o objetivo de preservar o tamanho e formato de cada célula individualmente, além de superar a limitação de dois métodos já conhecidos da literatura: os algoritmos de morfologia de Watershed e granulometria, que foram comparados e demonstrou que a precisão e a correlação do método proposto ultrapassaram os dois outros métodos mencionados anteriormente.

Wei *et al.* (2015) desenvolveram um método automático para detectar e contar as hemácias em imagens em lâminas de sangue. O artigo possui o detalhamento de cada etapa com imagens, gráficos e equações, que começam na etapa de pré-processamento, que retiram as outras células (como WBC e plaquetas). Depois, segue para etapa de classificação, que escolhe área da região fechada, separando as células em individuais ou aglomeradas pelo critério de circularidade, sendo usado o algoritmo Watershed para contar as células individuais e o algoritmo K-means para contar e melhorar a eficiência e precisão da segmentação das células aglomeradas. Em seguida, ocorre a contagem das hemácias individuais, sendo que essa contagem foi baseada no típico formato bicôncavo das RBCs e os autores ressaltam que o tempo é menor de processamento por possuir menos cálculos. Depois de retirar essas hemácias, restam somente as aglomeradas, criando uma sub imagem, que são contadas por outro algoritmo, como apresentado anteriormente. Ao final, a contagem de cada grupo de cada tipo de célula será somada resultando no total de hemácias, com tabelas que demonstram a precisão alcançada.

Abu-Qasmieh (2018) organizou um novo método automatizado para a separação e segmentação de RBCs aglomeradas ou sobrepostas, através de algoritmos que identificam o formato elíptico e a extração do ponto côncavo. Além de fazer a comparação entre os dois algoritmos, *Iterative Random Hough Transformation*

(IRHT) e *Direct Least Squares Fitting* (DLSF), foram avaliadas as velocidades e as precisões na detecção de objetos elípticos, sendo que o algoritmo DLSF teve melhor desempenho. Primeiro, o algoritmo tinha que testar se o objeto está distorcido ou não obtendo primeiro a borda da região utilizando o algoritmo de detecção de borda Canny. Em seguida, ocorrem várias operações morfológicas com o objetivo de detectar os pontos côncavos que são procedimentos necessários para segmentar objetos elípticos sobrepostos. Logo após, as células sobrepostas são separadas usando o novo método sugerido, que foi alterado no algoritmo DLSF, que sucede com procedimentos de separação, segmentação e detecção. O algoritmo testou diversos formatos como: células normais únicas, células sobrepostas, células distorcidas individuais e células bloqueadas pela borda da imagem, sendo possível identificar vinte elipses diferentes. A base de dados focou em imagens de lâmina de sangue de pessoas com transtorno de eliptocitose hereditária, fazendo uma análise quantitativa e estatística, sendo que os parâmetros foram as distribuições de tamanho das elipses (área), circularidade, excentricidade e elipticidade estimados além da contagem de RBCs. Os autores afirmam que mesmo que o algoritmo de transformação de Hough possa detectar várias instâncias de um modelo em uma única passagem, o consumo de tempo e a complexidade é muito mais alta comparada com DLSF que pode tratar os segmentos de contorno para cada candidato individualmente em um tempo muito menor. Além disso, no processamento dos segmentos de contorno de forma elíptica individualmente, a velocidade de reconhecimento em DLSF é mais alta do que em IRHT.

Sarrafzadeh *et al.* (2020) criaram um algoritmo responsável por detectar formatos circulares com a *Fast Circlet Transform* (FCT), sendo utilizado de três maneiras diferentes: detectar e contar RBCs; contar e localizar os glomérulos em imagens microscópicas de rins; e detectar parasitas em imagens microscópicas. Os autores classificaram os problemas em três categorias. Logo, três metodologias foram propostas para detectar padrões circulares: estimar o conjunto de círculos em uma imagem, localizando-os usando FCT e, em seguida, removendo os círculos de conflito; encontrar círculos através da análise direta dos coeficientes FCT; reconhecer círculos em uma imagem, modificando os coeficientes de FCT e reconstruir a imagem usando IFCT (FCT inverso). O primeiro é utilizado para quando os círculos que são apresentados na imagem estão próximos ou podem se sobrepor (como as RBCs). A segunda é para imagens que as estimativas não são simples e possuem cenários

razoavelmente complexos (não se encaixando nem na primeira e nem na terceira abordagem). A terceira possui pré-processamento adequado para quando não possui objetos sobrepostos e o fundo contém ruídos. Todos os métodos propostos foram comparados com outras abordagens e houve resultados melhores com as metodologias empregadas utilizando FCT.

Frejlichowski (2012) estruturou um algoritmo responsável por localizar e extrair os formatos das RBCs. O algoritmo é composto por três etapas: a imagem de entrada, que é em escala cinza, é transformada em binária utilizando um limite de histograma modificado para melhorar a qualidade em que se encontra. Logo após, cada RBC é localizada e extraída e finalmente algumas são rejeitadas por estarem no limite da imagem, portanto não podem ser usadas para reconhecimento ou para diagnóstico. Com isso, somente as RBCs que tiverem o tamanho completo foram extraídas das imagens. O algoritmo proposto foi comparado com o de outros autores e alcançou um patamar de qualidade média, ficando acima de 83% de eficiência de localização e detecção.

Savkare, Narote e Narote (2016) desenvolveram um algoritmo automatizado para a segmentação de células sanguíneas, mais especificamente as RBCs, sendo baseado em K-Mean Clustering. O algoritmo contém algumas etapas principais, começando pelo pré-processamento (que usa histograma em escala de cinza), para retirar o ruído e melhorar o contraste da imagem. Logo após, o passo da segmentação (usa o K-Mean clustering) e separação de células aglomeradas (Transformada de Watershed), finalizando com a contagem automática das células e avaliação dos resultados que alcançaram um geral de acurácia de 98,89%, demonstrando uma performance e eficiência alta.

Abbas (2015) preparou um método para contar RBCs através de um algoritmo feito no Matlab. O algoritmo consiste em separar os três componentes de cor, iniciando a segmentação escolhendo o componente verde (pois a imagem de RBC é mais clara) para fazer o histograma, assim observando a área que se for de determinado valor. Em seguida, será extraído o objeto e serem somados os objetos para a contagem de RBCs. O artigo só aborda a técnica, não sendo comparada com nenhum outro tipo e nem são utilizadas métricas de avaliação.

Shirazi *et al.* (2015) criaram um algoritmo responsável por melhorar a imagem e segmentação de RBCs. O algoritmo usa o filtro Wiener juntamente com a transformada Curvelet rápida discreta (FDCT) para aprimorar a imagem e fazer a

remoção de ruído. Seguindo, o algoritmo Snake e ortogonalização de Gram-Schmidt foram aplicados para detecção de limites e segmentação de imagens, respectivamente. O processo de reconhecimento das imagens automatizadas consiste em quatro etapas principais, incluindo pré-processamento (para resolver problemas de iluminação), segmentação de imagem, extração de características e classificação. Os autores afirmam que o sistema proposto se limita a pré-processamento e segmentação de imagens e que o principal desafio é a segmentação da célula de sangue, porque além de afetar fortemente o desempenho da classificação existem objetos complexos sobrepostos e variações de forma presentes na imagem. Os autores ressaltam que o método fornece resultados melhores na presença de fundos complexos e para imagens com formas e objetos sobrepostos. Além disso, eles concluem que a segmentação precisa da imagem é uma parte fundamental para a extração de características e classificação das células sanguíneas.

Qiu *et al.* (2020) desenvolveram um algoritmo baseado em rede neural convolucional da região (RCNN) para detectar e classificar RBCs, podendo ser saudáveis ou não saudáveis (falciforme), individuais ou múltiplas (sobrepostas ou encostando), com objetivo de ser *multi-label*. O algoritmo possui três etapas, sendo a primeira aplicar o “mais rápido” RCNN (Faster-RCNN) para extrair automaticamente uma única ou múltiplas células de uma imagem microscópica; em segundo lugar, foi implementado um ResNet pré-treinado para extração de características e desenvolver várias redes para obter o tipo de célula prevista; e, por último, foi utilizado o Gradient Boosting Classificador para determinar a presença de tipos de células “anormais”, sendo divididas em 6 subgrupos. Para a parte do algoritmo que aborda o Faster-RCNN, foram usadas duas métricas de avaliação: precisão média e a área sob a curva de recuperação de precisão, além do método proposto ser comparado com dois métodos que são de segmentação (Watershed e U-Net), demonstrando que o novo método proposto consegue obter resultados melhores e mais precisos.

Hortinela *et al.* (2020) estruturaram um algoritmo para a detecção, classificação e contagem de RBCs. O algoritmo possui a primeira etapa que utiliza o algoritmo de Watershed para melhorar a imagem em áreas específicas com células segmentadas e com a estrutura de Máquina de vetores de suporte (SVM), o sistema foi treinado e considerou parâmetros como: área, perímetro, diâmetro, formato, entre outros. Para identificar células que são classificados em: normais, equinócitos, eliptócitos, dacrócitos, esferócitos, célula-alvo e estomatócitos. Os resultados

mostraram que o sistema tem uma alta precisão na classificação de diferentes tipos de células anormais. E os autores evidenciam que é possível melhorar o sistema ainda mais com o uso de outras técnicas de processamento de imagem para comparar e analisar as células. Além disso, a utilização de câmeras de alta resolução diminui os erros que são feitos pelo uso de imagens de baixa qualidade.

Bailo *et al.* (2019) desenvolveram um algoritmo baseado em rede adversária geradora (GAN) para aumentar a quantidade de imagens de sangue (RBC), mantendo o fotorrealismo para que seja possível treinar melhor os algoritmos quando as bases de dados possuem tamanho pequeno. Os autores relatam que com o aumento das imagens, ocorre uma melhora na precisão dos algoritmos em detectar e classificar as células para aumentar o desempenho. O algoritmo gera imagens de RBCs por meio da incorporação de GAN condicional e máscara de segmentação sintética, começando com o conjunto de dados que é utilizado para treinar o GAN para realizar a tradução de uma máscara de segmentação para uma imagem de sangue e para construir a célula do banco de dados de exemplos de formas. Após o treinamento da rede, é gerado automaticamente diversas máscaras de segmentação de instância em execução, e criar um número definido de várias amostras de dados que são posteriormente usados como dados de treinamento adicionais para segmentação e detecção, sendo que aumento da quantidade das imagens inclui rotação, escala, inversão horizontal e vertical. Para a avaliação, a precisão média (AP) é usada como uma métrica de precisão.

Li *et al.* (2021a) elaboraram um *software* para um sistema de imagens sem lentes, sendo substituída a medição do raio RBC por uma medição do raio do anel de difração, podendo melhorar a resolução da imagem por ser maior criando uma proporção. O sistema funciona coletando as sombras (que são, na verdade, franja de difração do objeto) das células que passam por cima de um CIS, observando informações como quantidade, tamanho, formato, entre outros. Os autores relatam as duas dificuldades que precisam ser ultrapassadas: efeito de difração e a resolução baixa, sendo usado o raio de anel de difração para contornar tais problemas. O algoritmo demonstra uma relação entre a difração padrão e o tamanho de RBC, que pode ser usado para medir os diferentes tamanhos das hemácias, sendo considerado que a borda de um RBC seja redonda, o algoritmo coleta a difração de um ponto arbitrário na borda de um RBC que pode ser considerado como a difração na linha

tangente desse ponto, sendo utilizado a Difração de Fresnel em uma borda reta para descrever a difração de RBCs.

Os 22 artigos apresentados possuem técnicas diferentes entre si. Alguns focam em identificar RBCs saudáveis e outros as não-saudáveis, também podendo destacar células individuais ou sobrepostas. Porém, todos possuem o mesmo objetivo de tentar alcançar a melhor precisão na identificação e classificação das RBCs, sendo considerados relevantes para a pesquisa. No entanto, tais artigos são focados somente em RBCs e os obstáculos que podem ser encontrados como células aglomeradas. Nesse sentido, este trabalho vai expandir a parte da pesquisa de identificação e classificação para as outras células sanguíneas, que são as WBCs e as plaquetas.

2.2.4 Técnicas para Imagens de WBCs

No Quadro 6 estão exibidos os artigos que demonstram métodos para identificar as WBCs em imagens microscópicas. Os primeiros focaram em separar as WBCs aglomeradas ou sobrepostas (NILSSON; HEYDEN, 2005; ZAMANI; SAFABAKHSH, 2006). Os seguintes conseguem reconhecer as WBCs através do formato (RAY, 2010; SHIRAZI *et al.*, 2016) e da cor (AL-DULAIMI *et al.*, 2019; MOHAMMED; DAWAY; JOUDA, 2020; TOMCZAK *et al.*, 2021). Alguns focaram em WBCs saudáveis e não-saudáveis (MORADIAMIN *et al.*, 2016; SHAHIN *et al.*, 2017) ou em classificar em subgrupos menores (TOZETTI *et al.*, 2015; ASHOUR; WAHBA; GHANNAM, 2021) do que todas as cinco WBCs que existem (neutrófilos, eosinófilos, basófilos, monócitos e linfócitos). Outros autores destacaram métodos híbridos (MOHAMED *et al.*, 2020) ou originais (SAYED; SOLYMAN; HASSANIEN, 2019; YANG *et al.*, 2021), até com construção de *hardware* em conjunto com o *software* (FANG *et al.*, 2017). Os próximos enfatizaram o uso com conjuntos diferentes de técnicas (WANG; MIN, 2006; PRINYAKUPT; PLUEMPITIWIRIYAWAJ, 2015; MOALLEM *et al.*, 2017; ZHONG *et al.*, 2019; KUMAR *et al.*, 2020), já os dois autores seguintes utilizaram VGG16 (WIJESINGHE *et al.*, 2020) e SVM (ABDULHAY *et al.*, 2018).

Quadro 6 - Visão Geral dos Artigos da Pesquisa 02 da seção Técnicas para Imagens de WBCs

Artigo	Tema/Categoria	Técnicas propostas	Técnicas comparadas	D	A	C
Nilsson e Heyden (2005)	Sangue - WBC (sobrepistas)	Morphological Techniques + Watershed	-	X	X	X
Zamani e Safabakhsh (2006)	Sangue - WBC (sobrepistas)	GVF + Snake	-	X	X	X
Ray et al. (2010)	Sangue - WBC (formato)	Morphological Techniques	-	X	X	X
Shirazi et al. (2016)	Sangue - WBC (formato)	FDCT + Morphological Techniques	-	X	X	X
Tomczak et al. (2021)	Sangue - WBC (cor)	Morphological Techniques	StarGAN/StarGANv2/Pix2PixHD	X	X	X
Mohammed, Daway e Jouda (2020)	Sangue - WBC (cor)	Morphological Techniques	BSV/IO/DE-based	X	X	X
Al-Dulaimi et al. (2019)	Sangue - WBC (cor)	Adapted JSEG Algorithm + Fuzzy C Mean	Morphological Techniques/ Detection's Borders/K-means Clustering/EM/FCM/NN Watershed/Watershed/ Chan-Vese Method/ Otsu's Threshold	X	X	X
Shahin et al. (2017)	Sangue - WBC (doença)	Morphological Techniques	-	X	X	X
MoradiAmin et al. (2016)	Sangue - WBC (câncer)	FCM + SVM + Morphological Techniques	-	X	X	X
Ashour, Wahba e Ghannam (2021)	Sangue - WBC (subgrupos)	CSRS	K-means/FCM	X	X	X
Tozetti et al. (2015)	Sangue - WBC (subgrupos)	CellProfiler (software)	-	X	X	X
Mohamed et al. (2020)	Sangue - WBC (híbrido)	Morphological Techniques + FCN	VGG16/VGG19/ResNet-50/DenseNet-121/DenseNet-169/InceptionV3/Inception-ResNet-V2/Xception/1.0 MobileNet224/Mobile NASNet-A	X	X	X
Sayed, Solyman e Hassanien (2019)	Sangue - WBC (original)	COFA	OFA/ MVO/ BA/ ALO/ MFO/ ABC	X	X	X

Quadro 6 - Visão Geral dos Artigos da Pesquisa 02 da Sessão Técnicas para Imagens de WBCs (Cont.)

Yang et al. (2021)	Sangue - WBC (original)	BSA-Net	VGG16/ ResNet50/ MobilenetV2/ InceptionV2/ DenseNet/ Xception/ basenet	X	X	X
Fang et al. (2017)	Sangue - WBC (hard)	Morphological Techniques	-	X	X	X
Moallem et al. (2017)	Sangue - WBC (conjunto)	Morphological Techniques	-	X	X	X
Wang e Min (2006)	Sangue - WBC (conjunto)	NDA + FCNN	TSMN/ Fuzzy Logic	X	X	X
Kumar et al. (2020)	Sangue - WBC (conjunto)	Morphological Techniques	-	X	X	X
Prinyakupt e Pluempitiwiriawej (2015)	Sangue - WBC (conjunto)	Linear e Naive Bayes	-	X	X	X
Zhong et al. (2019)	Sangue - WBC (conjunto)	Morphological Techniques	Combination Graph Segmentation/ Watershed Segmentation/ SVM/ Adaptive Histogram Segmentation	X	X	X
Wijesinghe et al. (2020)	Sangue - WBC (VGG16)	VGG16	-	X	X	X
Abdulhay et al. (2018)	Sangue - WBC (SVM)	SVM	-	X	X	X
Reena e Ameer (2020)	Sangue - WBC (deep)	DeepLabv3 + AlexNet	U-Net/ SegNet/ FCN- AlexNet	X	X	X
Xia et al. (2019)	Sangue - WBC (deep)	Faster R-CNN	-	X	X	X
Qin et al. (2018)	Sangue - WBC (deep)	Cell3Net	AlexNet/ Res-Net-18/ VGG16/ GoogleNet/ ZFNet/ DenseNet	X	X	X
Fang, Yu e Liu (2018)	Sangue - WBC (CNN)	CNN	-	X	X	X
Wang et al. (2019)	Sangue - WBC (CNN)	ANN + CNN	AlexNet/ GoogLeNet/ VGG	X	X	X
Jiao et al. (2018)	Sangue - WBC (CNN)	CNN	CNN	X	X	X
Habibzadeh, Krzyzak e Fevens (2013)	Sangue - WBC (CNN)	CNN	SVM	X	X	X

Quadro 6 - Visão Geral dos Artigos da Pesquisa 02 da Sessão Técnicas para Imagens de WBCs (Cont.)

Artigo	Tema/Categoria	Técnicas propostas	Técnicas comparadas	D	A	C
Fan et al. (2019)	Sangue - WBC (CNN)	LeukocyteMask	Watershed/FCN/U-Net	X	X	X
Malkawi et al. (2020)	Sangue - WBC (CNN)	CNN + (Softmax/ SVM/ KNN/ Random Forest)	-	X	X	X
Zhao et al. (2017)	Sangue - WBC (CNN)	CNN + Morphological Techniques	Method Seyed/HSVM	X	X	X
Yao et al. (2021)	Sangue - WBC (CNN)	TWO-DCNN	VGG19/ VGG16/ Inception-V3/ ResNet-50/ SVM/ MLP/ DT	X	X	X
Lalitha, Raghul e Premkumar (2020)	Sangue - WBC (CNN)	Densenet	-	X	X	X
Falcon-Ruiz, Paz-Viera e Sahli (2010)	Sangue - WBC (compactar)	Morphological Techniques	-	X	X	X

*D = Detalhamento das técnicas (Algoritmos)

*A = Aplicações microscópicas para células sanguíneas

*C = Classificação/Identificação de células sanguíneas

*deep = O autor usou técnicas com *deep learning*, não necessariamente sendo CNN.

Fonte: autoria própria.

Ainda em relação ao Quadro 6, os demais autores abordaram métodos com aprendizagem profunda (QIN *et al.*, 2018; XIA *et al.*, 2019; REENA; AMEER, 2020), e outros autores mais especificamente com o uso das CNNs (HABIBZADEH; KRZYZAK; FEVENS, 2013; ZHAO *et al.*, 2017; FANG; YU; LIU, 2018; JIAO *et al.*, 2018; FAN *et al.*, 2019; WANG *et al.*, 2019; LALITHA; RAGHUL; PREMKUMAR, 2020; MALKAWI *et al.*, 2020; YAO *et al.*, 2021). Por último, foi apresentada uma técnica para compactar as imagens (FALCON-RUIZ; PAZ-VIERA; SAHLI, 2010) sem perder as características para identificar as WBCs.

Nilsson e Heyden (2005) desenvolveram um algoritmo para segmentação de qualquer tipo de quantidade de WBCs sobrepostas em imagens de amostras de medula óssea, sendo possível separar até células em que os núcleos se encostam. O algoritmo começa fazendo uma segmentação excessiva da imagem em subpartes, para em seguida serem juntadas por um algoritmo de pesquisa rápida. O primeiro passo da segmentação é localizar o núcleo e o citoplasma das WBCs e depois separar pela transformada de Watershed. Após, a célula é montada consistindo em resolver

um problema de otimização combinatória baseado em modelo sujeito a restrições biológicas fundamentais. Os autores ressaltam que os resultados são promissores, sendo que algoritmo pode ser ajustado para outros tipos de células sobrepostas e que os resultados atuais demonstram que o algoritmo consegue localizar cada célula corretamente na maioria dos casos.

Zamani e Safabakhsh (2006) elaboraram um algoritmo não-supervisionado baseado em Fluxo de gradiente vetorial (GVF), que é uma classe de forças externas para modelos de contorno ativos e com o algoritmo Snake para a segmentação de WBCs e separar de WBCs que as encostam ou que as aglomeram, usando a estratégia inicial do limite contornado pelo núcleo, pois na WBC a parte mais escura é o núcleo que pode ser localizado por uma análise de histograma adaptativa. Para obter o GVF Snake, o fluxo do vetor gradiente é usado como um campo de força externo, sendo usado para garantir que o contorno inicial conteria núcleo e o envolveria completamente. A imagem binária do núcleo, que é o resultado do limite do histograma, é estendida por operadores morfológicos que são dois operadores sucessivos: Dilatação e Erosão. Finalmente, os autores relatam que os resultados experimentais são eficientes.

Ray (2010) criou um algoritmo automático baseado em detectar WBC de formato de elipse em que são usados o ajuste da curva como um custo côncavo para o problema de minimização. O autor resalta que o algoritmo proposto pode ser estendido a um importante problema de visão computacional e análise de imagem chamado correspondência ou registro de recurso.

Shirazi *et al.* (2016) estruturaram um algoritmo para segmentação e classificação de todos os tipos de WBCs com base no tamanho e forma. O algoritmo inicia com o pré-processamento que retira os ruídos e transforma a imagem para ter alto contraste para melhorar o processamento, usando as técnicas de FDCT e filtro Wiener para aprimoramento de imagem. Após, é feita a parte de segmentação, que usa combinação de filtro de entropia, limiar e morfologia matemática para identificar as WBC. Depois, são extraídas as características das WBC e feita a classificação, sendo extraídos três tipos de características: geométricas, Matriz de Co-ocorrência de Nível de Cinza (GLCM) e Matriz de comprimento de execução de nível de cinza (GLRLM). Para encontrar o conjunto de características mais discriminante, é utilizado o método análise de componentes principais (PCA) e seleção sequencial para a frente (SFS) para selecionar as características, chegando em 20 conjuntos de recursos de

um conjunto de 86, sendo passado para a rede neural para classificação, que são as redes neurais de retro propagação (BPNN) que são usadas para o reconhecimento de WBC em seus cinco tipos. Os autores ressaltam que os resultados são positivos, sendo superado o problema da sobreposição de células e que atingiram 100%, 96,15%, 92,30%, 92,30% e 96,15% de precisão para basófilos, eosinófilos, monócitos, linfócitos e neutrófilos, respectivamente.

Tomczak *et al.* (2021) desenvolveram um algoritmo para "colorir" de forma digital as imagens de WBC não coloridas e classificando-as. O algoritmo tem como objetivo classificar automaticamente imagens não "coloridas" e "colorir" digitalmente para o profissional da saúde saber distinguir. Os autores relatam as contribuições do método proposto que são: apresentação de uma nova combinação de geração de imagem, como segmentação e reconstrução e demonstrar sua influência sobre a qualidade das imagens geradas, tanto em termos de granulação, finos detalhes e estruturas bem definidas; realizar avaliações quantitativas e qualitativas completas, comparando o método proposto aos métodos de ponta com estudos exaustivos que discutam as deficiências e possíveis melhorias. O algoritmo consiste em coloração digital e adaptação de domínio, sendo fundamental transferir a estrutura das células que são utilizadas pelo patologista como pontos de referência, como núcleo, citoplasma e granularidade para a tarefa de classificação. Assim, a diversidade dos resultados obtidos não é tão importante quanto a preservação da estrutura. O segundo objetivo é adaptação de domínio, como criar uma característica invariante de domínio que faça a representação, de modo que, por exemplo, imagens não manchadas sejam classificadas corretamente dados os marcadores fracos de outros domínios de imagens manchadas. Para a avaliação do método proposto, foi comparado a outros métodos de adaptação de domínio em termos de classificação precisão, como StarGAN, Star- GANv2 e Pix2PixHD, sendo métodos de última geração em termos de tradução de imagem para imagem e classificação multidomínio, fazendo avaliações quantitativas e comparando as imagens com os resultados de tradução de diferentes métodos.

Mohammed, Daway e Jouda (2020) criaram um algoritmo responsável por detectar automaticamente WBCs utilizando o componente de cor, sendo usado o Matlab. O algoritmo começa com as imagens coloridas, transforma no tamanho correto, faz a conversão binária de cores, emprega um filtro mediano e por fim, faz a detecção da região que contém a WBC. A avaliação foi feita entre a acurácia da

contagem manual e automática dependendo da taxa de detecção e taxa de alarme falso, e também foi comparada a eficiência da detecção com outros métodos: método de vetores de suporte de limite (BSVs), o método de algoritmo de Wang, o algoritmo Iterative Otsu (IO) e o método baseado em algoritmo genético. Os resultados demonstraram que o método proposto conseguiu alcançar 100% de taxa de detecção, além de ser preciso, robusto e apresentar estabilidade comparada a outros métodos.

Al-Dulaimi *et al.* (2019) estruturaram um algoritmo completamente automatizado para segmentação de WBCs. O algoritmo utiliza JSEG baseado na distribuição de cor-textura e usando o Fuzzy C Mean para segmentar e contar WBCs e seus núcleos (três tipos principais e sete subtipos). O algoritmo é um método não supervisionado usado para segmentação e classificação que possui três etapas. A primeira é que cada região na imagem WBC contém uniformemente padrão distribuído de cor e textura. A segunda é que as informações de cor em cada região da imagem WBC podem representar algumas cores quantizadas. A última é que as cores entre duas regiões vizinhas são distinguíveis, como parede celular e núcleo. O método proposto foi avaliado usando quatro índices: Jac- Erro de distância do cartão, Índice Rand, Deslocamento de limite erro de Ment e F-Index, além de ser comparado com 13 outros métodos de segmentação: Operadores morfológicos, detecção de bordas e limites, K-means baseados no espaço de cores, clustering, modelo de detecção de região, Fuzzy C Mean (FCM), método de aprendizagem auto supervisionado, limiar duplo, rede neural baseada em Watershed, agrupamento K-Means e algoritmo de Watershed, método de Chan-Vese e Otsu's threshold. Os resultados demonstraram que o método proposto apresentou um desempenho superior ao dos outros métodos, pois mostrou melhor habilidade para detectar WBCs e seus núcleos, e não falharam em produzir bons resultados quando têm bordas complexas, células sobrepostas, baixo contraste ou distorção de cor ocorrem entre o WBC e seu núcleo.

Shahin *et al.* (2017) elaboraram um algoritmo capaz de identificar e classificar as WBCs saudáveis e não saudáveis usando medidas de similaridade de multiescala baseada no domínio neutrosófico. O artigo ressalta que as principais contribuições são que o sistema proporciona uma segmentação totalmente automatizada, capaz de contar e medir cada estrutura das WBCs, com precisão segmentar de cada WBC a um núcleo e um citoplasma, separando em leucócitos saudáveis (neutrófilos, eosinófilos, basófilos, monócitos, linfócito) e células não saudáveis (células blásticas conectadas, única célula conectada), aplicando os conjuntos neutrosóficos a um novo

domínio das imagens e modificando medida de pontuação de similaridade para ser adaptável com multiescala e com multicritérios. O método proposto considera as informações de cor de WBCs, a semelhança entre diferentes componentes de cor é medida com base em pontuação de similaridade de conjunto neutrosófico. A medida Neutrosófico de pontuação de similaridade (NSS) anterior dependia de critérios como a intensidade, a homogeneidade, o local e os critérios de intensidade média. As imagens usadas para extrair as WBCs possuem resoluções, contrastes, iluminações e fontes diferentes. O desempenho do sistema é avaliado com métricas qualitativas e quantitativos e com os resultados apresentando altas precisões dos segmentos quantitativos. Além disso, o método proposto é adaptável com diferentes resoluções ou condições de luz.

MoradiAmin *et al.* (2016) desenvolveram um método para detecção auxiliada por computador, com foco em conjunto de geométricas e características estatísticas com núcleo da célula acompanhada com PCA e um classificador de conjunto SVM para colocar em subtipos de células cancerosas. O método proposto tem os seguintes passos: aquisição e pré-processamento (equalização do histograma) de imagens, segmentação do núcleo da célula, pós-processamento e extração das características (FCM, geométrica e textura) da imagem de núcleo, análise PCA das características do espaço e classificação usando SVM (linear, quadrático, polinomial, funções de base radial (RBF), perceptron multicamadas (MLP) e calculou a média) para classificar em não-cancerígena, L1, L2 ou L3. A avaliação foi feita com as métricas de sensibilidade, especificidade, acurácia e precisão. Os autores ressaltam que um problema encontrado foi a ausência de dados disponíveis publicamente, não sendo possível comparar diretamente com outros métodos propostos, entretanto nas métricas avaliadas, os resultados demonstraram alto desempenho.

Ashour, Wahba e Ghannam (2021) estruturaram um algoritmo de classificação-segmentação em cascata proposta reversível (CSRS) de detecção automática, contagem e reconhecimento de WBC, mais especificamente em basófilos e eosinófilos em imagens microscópicas. O algoritmo possui as etapas em que começam por determinar a localização, avaliando a cor diferente entre espaços, segmentar, extrair e capturar as características e treinar os sistemas de classificação para definir seus parâmetros para o melhor desempenho. O método proposto também foi comparado com outras técnicas de segmentação, com o algoritmo K-means e o algoritmo FCM com diferentes números de centros de cluster, sendo que em

comparação ao sistema proposto tem um desempenho melhor em termos de tarefas de segmentação, contagem e classificação. Os autores ressaltam que os principais desafios com o projeto de qualquer computador assistido com sistema de detecção e contagem de imagens de sangue é a segmentação eficiente e única para abordar diferentes tipos de WBCs, seguido pela complexidade no processo de classificação devido à presença do fundo, o que adiciona ruído às características extraídas.

Tozetti *et al.* (2015) desenvolveram um método para contagem automatizada de WBC usando microscópio e classificando em subgrupos (linfócitos, monócitos e neutrófilos). O método utilizou-se do software CellProfiler para analisar e classificar as imagens das células sendo comparada a contagem manual feita em laboratórios, obtendo um resultado muito próximo. Para relacionar os resultados, uma regressão linear foi realizada para a contagem das subpopulações e para a contagem de leucócitos por mL.

Mohamed *et al.* (2020) elaboraram um algoritmo híbrido de duas etapas para detecção e classificação de WBC com alta precisão. A primeira etapa é usar um modelo de pré-treinamento para extração de características. A segunda etapa é captar essas características para um classificador para melhorar a precisão e a Rede totalmente conectada (FCN) é colocada em cima do pré-treinado estágio de extração. Os autores fizeram testes para avaliar qual o melhor modelo, sendo vistas dez estruturas diferentes e testados seis classificadores distintos. O algoritmo é resumido em pré-processamento de imagem, extração de características, classificação e avaliação de desempenho. Dez modelos de aprendizagem profunda pré-treinados são usados de forma independente como extratores de características em imagens segmentadas corretamente, sendo oito modelos principais: VGG16, VGG19, ResNet-50, DenseNet-121, DenseNet-169, InceptionV3, Inception-ResNet-V2 e Xception e dois modelos recentes com arquitetura leve: 1.0 MobileNet224 e Mobile NASNet-A. Para avaliar o desempenho de cada combinação, são usadas as métricas de exatidão, precisão, sensibilidade, especificidade, pontuação F1, precisão do classificador, entre outros. Através de tais testes, observaram que o modelo 1.0 MobileNet-224 e LDA com DenseNet-169 têm 98% como o valor mais alto, sendo esses dois híbridos considerados os melhores para o conjunto de dados proposto. Comparando com outros métodos propostos por outros autores, é possível observar que o híbrido conseguiu resultados melhores na maioria das métricas de classificação.

Sayed, Solyman e Hassanien (2019) criaram um algoritmo baseado em *Chaotic Optimal Foraging Algorithm* (COFA), sendo a combinação de caos com meta-heurística. O caos é considerado uma simulação de sistemas não lineares, os quais interagem de maneira dinâmica, sendo caracterizado por três propriedades: propriedade semi-estocástica, afetabilidade contra as condições iniciais e ergodicidade. Já a meta-heurística tem dois processos principais que são responsáveis por explorar, otimizar e encontrar a solução mais rápida. O artigo mostra que o desempenho do COFA apresentado foi avaliado em respeito à exploração, a trajetória do *Optimal Foraging Algorithm* (OFA), histórico de pesquisa, melhoria da aptidão da população e taxa de convergência. Os resultados mostraram que o método proposto supera outros algoritmos conhecidos da literatura (OFA, *Multiverse Optimization* (MVO), *Bat Algorithm* (BA), *Ant Lion Optimizer* (ALO), *Moth Flame Optimization* (MFO) e *Artificial Bee Colony* (ABC)) e foi usado para detecção de leucócitos, incluindo leucemia linfoblástica, mostrando que os resultados são quantitativamente e qualitativamente superiores.

Yang *et al.* (2021) desenvolveram um novo algoritmo automático de rede neural para identificação de WBCs em subtipos (eosinófilos, linfócitos, monócitos e neutrófilos), chamada de Rede Bilinear de Autoatendimento (BSA-Net). O algoritmo combina mecanismo de atenção (pode capturar dependências de longo alcance) com estratégia bilinear (pode reter interações de diferentes características de saída). O algoritmo não usa tecnologia de segmentação, usa o BSA-Net para ter atenção para aumentar o foco nos principais locais e menos no fundo, para poder usar a estratégia bilinear para capturar a interação entre características. Os autores afirmam que o método proposto possui quatro componentes principais: a rede de extração de características, modelo de atenção, normalização e classificação. A avaliação foi feita através das métricas: exatidão, precisão, sensibilidade e pontuação F1, além de ser comparado com outros métodos, Tendo o método proposto apresentado melhores resultados.

Fang *et al.* (2017) estruturaram um equipamento sem lentes baseado na técnica de sombra para classificação automatizada de WBC, sendo separado em três subgrupos (neutrófilos, monócitos e linfócitos). O aparelho é composto por três partes: um sensor CMOS, LED e um *software* de processamento de imagem. O *software* é também composto por três partes: algoritmo de segmentação automática, o algoritmo de extração de vetor de características de Distribuição Projetada da Borda Principal

(PPED), que foi desenvolvido para uso no sistema de reconhecimento de imagem *hardware* baseado em processadores neurais associativos, e o algoritmo de classificação de células. Os autores afirmam que, mesmo que o desempenho ainda tenha sido menor do que os métodos tradicionais, foi suficiente para demonstrar que o instrumento proposto e o método de classificação são eficazes. Além disso, é ressaltado que é até possível criar uma tecnologia em escala de *chip* para a criação de um instrumento ainda menor, sendo possível ampliar a aplicação de equipamentos que possibilitam um método mais conveniente de exame de sangue de rotina.

Moallem *et al.* (2017) elaboraram um algoritmo para ser utilizado em um *smartphone*, que usa ML e análise de imagem para identificar as RBCs infectadas. O objetivo foi obter mais informações nas RBCs e como a doença da malária afeta as mesmas. Porém, é necessário que tenha a eliminação das WBCs para que não afete na contagem das RBCs. O sistema consegue identificar 5 diferentes WBCs, fazer a detecção e a segmentação, tendo três etapas fundamentais: pré-processamento para extrair a região de interesse; uma etapa de detecção de células usando um método de filtragem de intervalo e uma etapa de segmentação de células utilizando o conjunto de diferentes contornos. Na etapa de pré-processamento, a região de interesse (ROI) na imagem é extraída. A etapa de detecção de células localiza qualquer WBC presente na imagem usando uma versão filtrada. E na última etapa, de segmentação celular, os leucócitos detectados são segmentados utilizando um método de definição de nível. O artigo mostra imagens e tabelas que demonstram a precisão do algoritmo proposto em segmentar WBCs com diferentes formas e tamanhos e diversos tons de coloração. Além de lidar com várias propriedades morfológicas de texturas de núcleo e citoplasma.

Wang e Min (2006) desenvolveram um algoritmo de detecção (NDA) baseado em Rede Neural Celular Difusa (FCNN). O algoritmo aproveita as vantagens de outros dois algoritmos: a segmentação de limiar seguida por morfologia matemática (TSMM) e o método de Lógica Fuzzy. A combinação faz com que o NDA consiga detectar quase todas as WBCs e o processo é composto por quatro etapas: aquisição, detecção, extração de característica e classificação. No artigo, são mostrados fluxogramas, equações e figuras que demonstram o passo a passo e também detalham as vantagens e desvantagens dos dois algoritmos que basearam a NDA. Finalizando, as imagens e as tabelas comprovam que o algoritmo proposto possui melhor performance.

Kumar *et al.* (2020) criaram um algoritmo responsável por segmentar e encontrar o núcleo de WBCs, sendo avaliada a eficiência dos algoritmos de detecção de borda (Log & Canny), K-means, segmentação baseada em cores, método transformado linear e método de limite usando o Matlab. Os autores avaliaram os diferentes algoritmos de segmentação de imagem e compararam os resultados de cada algoritmo separadamente. A conclusão é que o método Canny & Log fornece a melhor segmentação em algoritmos de segmentação baseados em borda. Os autores revelam que cada algoritmo de segmentação tem certos parâmetros, como Edge limiar, limiar do algoritmo K-means de interesse do contexto etc. e um resultado de algoritmo bem segmentado tem diferentes valores dos parâmetros da imagem. Os autores ressaltam que é complexo afirmar qual algoritmo é mais eficiente na parte de segmentação do que outro, seja para uma imagem ou coleção de imagens, ou geralmente, toda uma classe de imagens. Além dos autores afirmarem que o Matlab armazena a maioria das imagens como matrizes bidimensionais, que combinando cálculo, visualização e programação em um ambiente fácil de usar, a qual mostra os resultados de forma mais didática e eficiente.

Prinyakupt e Pluempitiwiriyawej (2015) desenvolveram um método de segmentação para WBC com o objetivo de classificar entre os cinco tipos existentes. O sistema proposto é separado em cinco etapas principais: pré-processamento, segmentação (núcleo e região do citoplasma), extração de características, seleção e classificação de características. Os autores ressaltam que os principais problemas encontrados na segmentação WBC são sua variedade de formas e tamanhos e afirmam que cada WBC tem um núcleo, citoplasma e alguns grânulos. Após as características serem extraídas, o SFS é aplicado para selecionar um subconjunto de recursos sem qualquer transformação. E, por último, os classificadores Linear e Naive Bayes são empregados para classificar as células. O sistema foi testado por dois conjuntos de dados que apresentam resoluções diferentes e foram avaliados com as métricas de acurácia, sensibilidade, especificidade e precisão, sendo que ambos os conjuntos de dados apresentaram mais de 90% de precisão. Demonstrou-se, assim, ser um método rápido e eficiente e também que o processo de segmentação proposto pode ser calibrado para realizar diferentes tamanhos ou formatos de imagem, desde que a resolução seja conhecida.

Zhong *et al.* (2019) criaram um algoritmo para segmentar as WBCs, aplicando duas técnicas diferentes: restrição de esparsidade e restrição de geometria. O artigo

explica que a primeira aumenta os recursos da imagem para segmentação nuclear precisa. Já a segunda ajusta para completar o limite parcial da célula retirada da imagem original. As principais contribuições são que a imagem esparsa consegue destacar as características importantes e esmaecer as irrelevantes. A estratégia do modelo também consegue detectar limites de células incompletas (inspirada em *Random Sample Consensus-RANSAC*), além de resultados experimentais comprovarem a alta precisão do algoritmo implementado. O método proposto foi comparado com outros métodos conhecidos como: *Combination Graph Segmentation*, *Watershed Segmentation*, SVM e *Adaptive Histogram Segmentation*. Com os resultados, foi possível observar que o método proposto conseguiu apresentar melhores valores de precisão de segmentação, utilizando dois conjuntos de dados que incluem ambas as imagens de coloração rápida e padrão, comprovando a eficácia.

Wijesinghe *et al.* (2020) elaboraram um sistema completamente automatizado para detecção e classificação de WBC em seus respectivos tipos. O algoritmo apresenta uma detecção e classificação (VGG16) com precisões altas, além de ser capaz de extrair WBC de imagens com várias células, e a capacidade de filtrar partículas de manchas, RBC nucleado e células sobrepostas que outros tipos de métodos não conseguem fazer. O algoritmo se utilizou de imagens de sangue caninas, porém os autores afirmam que é facilmente calibrado para amostras humanas, pois o sistema é o que fornece a capacidade de processar imagens microscópicas brutas obtidas através de um sistema de aquisição automaticamente. O sistema proposto consiste em três etapas principais: pré-processamento, detecção e localização de WBC e classificação. O classificador é um modelo VGG16 que foi pré-treinado e é usado para prever as células em 5 classes WBC, cujas camadas são totalmente conectadas no modelo e são alteradas de acordo com a tarefa de classificação, sendo que também os parâmetros do modelo personalizado são ajustados usando o conjunto de dados ImageNet como peso inicial. No artigo, são mostradas as tabelas com as acurácias alcançadas, sendo um modelo com alto desempenho.

Abdulhay *et al.* (2018) estruturaram um algoritmo responsável por realizar a segmentação automática de WBC (benigno e maligno) usando imagens microscópicas estáticas, sendo feito através de três principais componentes: melhoramento da imagem, SVM para a segmentação e filtrando ROI baseado em padrões binários locais e de características de textura. Antes de começar a

classificação, ocorre a utilização de filtros para minimizar ruídos e o treinamento do algoritmo. Para testar a eficácia, foi avaliado em diferentes estágios, sendo eles: a exatidão da segmentação, comparação das medições usuais com leituras manuais que são gerados por especialistas e a exatidão das medidas habituais para enfermarias que reconhecem os casos que já possuem rótulo. Além disso, foi utilizado o algoritmo K-fold de avaliação cruzada para qualificar a precisão, a sensibilidade e a especificidade. A precisão da identificação na técnica utilizada é de cerca de 95,3%, com sensibilidade de 100 e 91,66% de especificidade.

Reena e Ameer (2020) desenvolveram um algoritmo baseado em *deep learning* para segmentar e classificar WBCs. Para a segmentação o algoritmo é desenvolvido é o DeepLabv3 e para a classificação é desenvolvido o algoritmo AlexNet para classificar cinco tipos de leucócitos do sangue. O algoritmo faz a combinação de segmentação semântica (alcançar melhor precisão em localizar a WBC, tendo assim em uma melhor precisão) e a classificação é baseada na transferência da aprendizagem. Para cada etapa, é demonstrado o passo a passo feito por cada parte do algoritmo, com imagens que ilustram como as células foram identificadas. Com gráficos que mostram as matrizes de confusão que confirmam que os resultados foram precisos. A precisão obtida para a segmentação e classificação dos cinco tipos de WBC na fase de teste é maior que 99% para todos os tipos de células, exceto linfócitos e monócitos. Além disso, a classificação possui a precisão de 100% para basófilos e eosinófilos, enquanto para o restante células, a precisão é superior a 98%. Porém, para avaliar de fato o desempenho da técnica proposta, foi comparado com os resultados obtidos usando modelos de segmentação semântica, como U-Net, SegNet e FCNAlexNet. As métricas para avaliação de desempenho foram precisão, especificidade e a pontuação. O algoritmo proposto produz uma segmentação com acurácia média de 98,22% e uma precisão de classificação de $98,87 \pm 1\%$, que é superior aos outros métodos existentes.

Xia *et al.* (2019) elaboraram um algoritmo para detecção de WBCs usando DNN, sendo baseado em RCNN mais rápidos (Faster RCNNs). O algoritmo possui Redes de propostas regionais (RPN) que gera várias regiões com o potencial de se obter o objeto desejado, tendo um classificador e também um regressor. Após que a região é identificada, é passado para a rede de detecção que reconhece qual classe o objeto pertence, sendo em conjunto com um processo de aprendizagem por transferência para aplicar o método proposto ao microscópio com o objetivo de

conseguir fazer a detecção de células sanguíneas. Os autores ressaltam que é, potencialmente, uma nova técnica de contagem de células sanguíneas em dispositivos micro fluídicos para o serviço de Ponto de atendimento (POC), com resultados e performance ótimos, com chances de ser comercialmente disponível, mesmo que demore para ser usado em práticas de clínicas.

Qin *et al.* (2018) criaram um algoritmo (Cell3Net) para classificar de forma mais refinada as WBCs através de *deep residual learning*. O processo para o desenvolvimento começa com adquirir amostras sanguíneas de hospitais locais e criar os bancos de dados de WBC para treinamento e teste de performance, aumentar a quantidade de imagens, desenvolver o algoritmo com o uso de CNN para extrair o formato da célula. Em seguida, é feito o treinamento com múltiplas estratégias, com a seleção e combinação para fazer o classificador obter um melhor desempenho (40 categorias). E, por fim, uma tabela de comparação entre o método proposto e outros métodos (AlexNet, Res-Net-18, VGG16, GoogleNet, ZFNet, DenseNet) demonstra que a performance é melhor alcançada com Cell3Net. Os autores abordam as principais contribuições que são: a arquitetura *neural network* que classifica as células; os treinamentos com estratégias combinadas e as amostras "originais". Os autores afirmam que o algoritmo precisa de melhorias, mas o Cell3Net já alcança ótimos resultados (percentualmente) nas métricas de classificação.

Fang, Yu e Liu (2018) estruturaram um algoritmo baseado em CNN (LeNet-5) em um sistema de imagens que não contém lentes para a segmentação de WBC, criando três grupos: linfócitos, monócitos e neutrófilos. O algoritmo CNN foi treinado e testado pelo MATLAB e atingiu a conclusão de que tal método proposto é viável, alcançando uma acurácia de 90%. Porém, para aumentar a precisão, é preciso melhorar a resolução das imagens em cinza das WBC, tendo sido proposta pesquisa para ser feita no futuro.

Wang *et al.* (2019) desenvolveram um algoritmo CNN para detecção de WBCs em lentes objetivas de ampliação de 10x, sendo a primeira etapa para encontrar as regiões de interesse com Rede Neural Artificial (ANN) em nível de pixel baseado na cor e tamanho da WBC, com a segunda etapa responsável pela tarefa de classificação por arquitetura CNN, sendo usadas três arquiteturas de *deep learning*: Classificação ImageNet com redes neurais convolucionais profundas (AlexNet), indo mais fundo com as convoluções (GoogLeNet) e Redes convolucionais muito profundas para reconhecimento de imagens em grande escala (VGG), escolhendo a última por melhor

resultado na classificação. A avaliação foi feita pelas métricas de precisão, *recall*, exatidão e formulação, que apresentaram ótimos resultados de desempenho nas métricas de classificação.

Jiao *et al.* (2018) criaram um algoritmo baseado em CNN otimizada para detectar macrófagos CD68 em conjunto com WBCs, com o foco em empregar a CNN para detectar WBCs e aumentar os resultados da segmentação. O algoritmo possui máscaras de tamanho diferentes que, com base no resultado da classificação CNN, são usadas para pós-processar os resultados de segmentação obtidos pelo LIOTsu que é método de limiarização. As máscaras usadas no algoritmo fornecem informações sobre localização, conectividade, *clustering*, distribuição e excentricidade. No artigo ocorre a comparação com outras arquiteturas de CNNs em termos de precisão, sensibilidade e acurácia e os resultados experimentais confirmam que a arquitetura CNN proposta atinge a melhor precisão na classificação de células CD68-positivas em comparação com outras arquiteturas CNN.

Habibzadeh, Krzyzak e Fevens (2013) apresentaram a comparação de três sistemas que fazem a classificação automática de imagens de baixa resolução de WBCs em cinco grupos. A comparação é feita entre os algoritmos CNN com a arquitetura de LeNet5 com 8 camadas, SVM com recursos de intensidade e histograma e SVM com recursos de intensidade e histograma reduzidos em dimensionalidade pela análise de componentes principais do kernel (K-PCA). Antes de começar a classificação, o algoritmo determina aproximadamente a localização do núcleo de leucócitos e aumenta as bordas da WBC e, logo após, são usadas máscaras para remoção de ruído e para separar WBCs e RBCs em duas sub imagens individuais para separar as WBCs das RBCs. O artigo apresenta equações, imagens e tabelas para demonstrar todo o processo feito para alcançar os melhores resultados. E, para comprovar o melhor resultado entre os três algoritmos, foram utilizadas matrizes de confusão que, mesmo no caso de amostras ruins (imagens confusas, WBC pequenas e desbotadas), fizeram as contagens das WBCs serem muito mais precisas quando o classificador CNN é usado em vez de um classificador SVM.

Fan *et al.* (2019) desenvolveram um algoritmo chamado de LeukocyteMask, que é baseado em uma arquitetura Mask-RCNN aprimorada, para localizar e segmentar as WBCs. De forma automatizada, o algoritmo usa informações dos pixels para o treinamento de uma rede neural convolucional profunda que é empregada para localizar a ROI de leucócitos e, finalmente, a máscara de segmentação de leucócito é

obtido com base na ROI extraída por propagação direta da rede. Os autores afirmam existir algumas contribuições, sendo as principais que o LeukocyteMask é um método que segmenta leucócitos de uma maneira pixel a pixel, sendo primeiro localizado o WBC para em seguida segmentar e ocorrer comparações quantitativas e qualitativas entre o método proposto e os métodos atuais existentes. O algoritmo possui três etapas: extração de características, proposta de região e previsão. Não há intervenção manual e nem pré-processamento, sendo que o código "prevê" tanto "WBC" ou "não-WBC" (incluindo RBCs e ruído de fundo) e rotula para cada pixel na imagem e características como forma, cor e textura de células e informações espaciais. O artigo demonstra através de imagens como o algoritmo funciona, delimitando as áreas que são "supostamente" as WBCs com a máscara e cores, representando a confiança do pixel ser classificado como WBC. No artigo, houve a comparação de desempenho com diferentes técnicas de aumento de dados entre o método proposto com outros métodos existentes, como Watershed, FCN e U-Net. Tanto quantitativamente quanto qualitativamente, a abordagem feita por LeukocyteMasck alcança melhores resultados, não apenas com uma maior precisão de segmentação, mas também um desempenho mais estável.

Malkawi *et al.* (2020) estruturaram um algoritmo híbrido baseado em CNN (para extrair as características) combinado com outros algoritmos de ML como: Softmax (*default* do CNN), SVM, KNN e Random Forest (para serem os classificadores) para a classificação de WBCs recortados de imagens. O processo inicia com o pré-processamento e continua com o aumento da quantidade de imagens para treinar o algoritmo para, após ocorrer o treinamento, posteriormente extrair as características. Para avaliar a *performance* dos conjuntos de algoritmos foram utilizadas as métricas: acurácia, especificidade, sensibilidade e precisão. Os híbridos que apresentaram melhores resultados foram CNN-Random Forest e CNN-SVM comparados aos outros, demonstrando serem híbridos com alta *performance* e desempenho.

Zhao *et al.* (2017) criaram um algoritmo para detectar e classificar WBC automaticamente. O algoritmo é composto por uma primeira etapa que separa a WBC do fundo através de imagens microscópicas baseando-se na relação de cores e operações morfológicas. Logo após, é utilizado um recurso de granularidade (PRICoLBP) para cada WBC e, em seguida, o SVM é empregado para diferenciar eosinófilo e basófilo de outras WBCs. Para, finalmente, serem extraídas

características em alto nível com CNN de forma automática e usar Random Forest para reconhecer e melhorar a classificação das células em mais outros três tipos (neutrófilos, monócitos e linfócitos). O artigo afirma que o algoritmo proposto é diferente dos métodos tradicionais, sendo que as principais contribuições do método são: na etapa de detecção é distinto pois, em vez de converter a imagem colorida, foi feita a aplicação da relação simples entre os componentes Red e Blue; o algoritmo combina os núcleos para localização das WBCs; algumas células (eosinófilos e basófilos) possuem grânulos, que foram usados para classificar os outros tipos de células; para os outros tipos, foi usado CNN e Random Forest para a classificação com precisão; é um sistema de reconhecimento automatizado sem necessidade de operação manual. Os bancos de dados usados foram Cellavison, ALL-IDB e Jiashan que demonstraram que o sistema proposto possui melhor detecção e classificação em comparação com outros métodos (método Seyed e método hierárquico SVM (HSVM) e com menos tempo. Porém, os autores afirmam que não é 100% perfeito e que não consegue captar as WBCs em algumas situações específicas.

Yao *et al.* (2021) elaboraram um algoritmo com método de rede neural convolucional deformável otimizado ponderado em dois módulos (TWO-DCNN) para a classificação WBC (para base de dados com baixa resolução e com ruídos) e que calculou as características geométricas, ondas e de textura da WBC. O método proposto apresenta dois fatores principais. Um deles é a inicialização dos pesos característicos da ImageNet no primeiro módulo, um conjunto de dados WBC de alta qualidade que foi inserido no primeiro módulo para treinamento com a saída de pesos otimizados. O segundo fator está em que o próximo módulo foi otimizado pelos pesos transferidos para a classificação de conjunto de dados WBC de baixa qualidade. Para validar a robustez do método proposto, foi comparado com MLs clássicas de VGG19, VGG16, Inception-V3, ResNet-50, SVM, MLP e árvore de decisão (DT). As métricas para avaliação do desempenho foram: precisão, *recall* e pontuação F1 e área sob as curvas e *Receiver Operating Characteristic curves* (ROCs). O algoritmo demonstrou resultados melhores na robustez e na classificação WBC de baixa resolução e de dados ruidosos do que outros métodos.

Lalitha, Raghul e Premkumar (2020) desenvolveram um algoritmo para detectar e contar WBCs usando *deep learning* (CNN), sendo conhecido como Rede de convolução densamente conectada ou DenseNet. O algoritmo classifica os cinco tipos de WBCs existentes em polinuclear e mononuclear para identificação de

doenças. O artigo somente aborda a técnica e a explica, entretanto não compara com outros métodos e nem avalia com métricas.

Falcon-Ruiz, Paz-Viera e Sahli (2010) estruturaram uma técnica para compactar imagens de WBCs sem perder a qualidade, conhecido como JPEG2000, com o objetivo de descobrir a taxa de compressão (CR) máxima possível que não afete a identificação de WBCs. A estimação é feita por métricas de qualidades objetivas e a *performance* é avaliada usando diferentes algoritmos de segmentação. As WBCs são classificadas em cinco grupos, sendo necessário para diferenciá-los usar o formato, textura e cor. Por isso, é necessário manter as características e singularidades de cada célula para ser possível classificá-las. O resultado alcançado de CR foi de 142:1 como o limite para ser possível a identificação de WBCs.

Os 35 artigos utilizaram métodos para identificar e classificar as WBCs, desde técnicas originais até as que usam aprendizagem profunda. As técnicas são diversas, cada uma com sua característica para alcançar a melhor precisão, podendo focar ou não em células saudáveis, também podendo ser individuais ou sobrepostas, e que foram consideradas interessantes para a pesquisa.

2.2.5 Técnicas para Imagens de RBCs e WBCs

Conforme o Quadro 7, nesta seção são mostrados os artigos que abordam diferentes técnicas para identificar e classificar tanto as RBCs quanto as WBCs.

Entre os principais destaques, tem-se Loddo *et al.* (2017) que elaboraram um sistema automático para contar as WBCs e RBCs, também sendo possível contar células que estiverem afetadas por alguma doença ou apresentam uma estrutura atípica. Já Baghli *et al.* (2014) criaram um algoritmo híbrido para segmentação de imagens células sanguíneas (RBC e WBC) baseado na teoria da evidência. Alomari *et al.* (2014) também usaram métodos de segmentação, contudo o algoritmo usa o formato circular das RBCs e WBCs para detectar e contar de forma automática. Habibzadeh *et al.* (2011) também desenvolveram um conjunto de técnicas para contar RBCs e WBCs em imagens ruidosas. Zhang *et al.* (2019), por sua vez, desenvolveram um algoritmo fundamentado em YOLOv3 para contar RBCs, podendo ser individuais ou aglomeradas e WBCs individuais. E, conforme já citados na Pesquisa 01, Huang *et al.* (2016b) e Tom *et al.* (2019) aplicaram técnicas que abordam a super-resolução,

sendo que o primeiro apresenta também o desenvolvimento de um equipamento, além de haver a comparação de *softwares* diferentes e o segundo também faz uma proposta de método de algoritmo para a identificação de núcleos e citoplasmas.

Quadro 7 - Visão Geral dos Artigos da Pesquisa 02 da Sessão Técnica para Imagens de RBCs e WBCs

Artigo	Tema/Categoria	Técnicas propostas	Técnicas comparadas	D	A	C
Huang <i>et al.</i> (2016b)	Sangue - RBC e WBC	CNNSR	ELMSR	X	X	X
Loddo <i>et al.</i> (2017)	Sangue - RBC e WBC	CHT	-	X	X	X
Baghli <i>et al.</i> (2014)	Sangue - RBC e WBC	ESA	-	X	X	X
Alomari <i>et al.</i> (2014)	Sangue - RBC e WBC	<i>Morphological Techniques</i>	<i>CHT/ Morphological Techniques/WaterShed/ Color-based and Hough Transform</i>	X	X	X
Tom <i>et al.</i> (2019)	Sangue - RBC e WBC	SISR + CNN	-	X	X	-
Habibzadeh <i>et al.</i> (2011)	Sangue - RBC e WBC	<i>Morphological Techniques</i>	-	X	X	X
Zhang <i>et al.</i> (2019)	Sangue - RBC e WBC	YOLOv3	-	X	X	X

*D = Detalhamento das técnicas (Algoritmos)

*A = Aplicações microscópicas para células sanguíneas

*C = Classificação/Identificação de células sanguíneas

Fonte: autoria própria.

Observando os trabalhos realizados, destaca-se que Loddo *et al.* (2017) desenvolveram um sistema automático que é auxiliado por computadores que faz a detecção e identificação de WBCs e RBCs, sendo possível contar células que até estiverem afetadas por alguma doença ou apresentam uma morfologia anormal. O algoritmo começa colocando uma máscara na imagem e todas as células sobrepostas são analisadas separadamente, sendo usado parâmetros para a transformada circular de Hough (que procura objetos circulares), faz a segmentação usando ML, retirando regiões de interesse (WBCs, RBCs e regiões de plasma) e, por fim, conta as células que são identificadas pelo número de círculos detectados. Para a avaliação de desempenho de contagem, foram usados precisão, recall, medida F e taxa FNR de falso negativo a fim de destacar quando o algoritmo não é capaz de detectar uma célula presente na imagem. Os resultados demonstraram que para contagem de

WBCs e RBCs, obteve-se um valor médio de precisão de 99,2% para WBCs e 98% para RBCs, superando outros métodos conhecidos na literatura.

Baghli *et al.* (2014) estruturaram um algoritmo híbrido para segmentação de imagens células sanguíneas baseado na teoria da evidência, sendo o algoritmo de segmentação de evidências (ESA). O algoritmo faz a segmentação do núcleo e citoplasma (das WBCs) e das RBCs do fundo, sendo coloridas em cores diferentes para melhor observação. Sendo através de dois principais etapas: a teoria da evidência para classificar os pixels de uma determinada imagem transformada de Watershed e classificação evidencial (usando os *pixels* e colorindo os diferentes organismos). Os autores afirmam que a *performance* da segmentação foi boa o suficiente, entretanto ocorreram erros como alguns citoplasmas sendo classificados como RBCs. Os autores acreditam que isso ocorreu pelo fato que WBCs possuem diversos tamanhos e formatos, sendo, provavelmente, o que causa tal confusão.

Alomari *et al.* (2014) criaram uma técnica que faz a segmentação e contagem através de um algoritmo de detecção do formato circular para as RBCs e WBCs do sangue, sendo de forma automática em imagens de microscópio. O artigo apresenta fluxogramas sobre o objetivo geral proposto e depois específico para WBCs e RBCs. O artigo fez uma busca para obter o melhor algoritmo possível para casos de sobreposição ou de irregularidade de células, sem que precisasse de muito processamento ou de muita memória consumida, com mais fatores como número de *pixels* e raio da célula. Tudo isso para ocorrer a melhor forma de detecção de círculos que possui um limite mínimo e máximo para detectar as bordas, criando um limiar de espessura para identificar a célula, uma vez que existem células com formato irregular. No artigo, existem figuras que demonstram o processo feito, as equações usadas para a melhor identificação, tabelas com resultados com precisões em porcentagens e com uma tabela final que compara o método proposto e por outros autores, quais as porcentagens alcançadas pela precisão, mostrando que o algoritmo desenvolvido apresenta ótimos resultados. Mesmo não apresentando um algoritmo de super-resolução, esse é de grande interesse para esta pesquisa que conta com a classificação de células.

Habibzadeh *et al.* (2011) desenvolveram um sistema com o objetivo principal de contar RBCs e WBCs em imagens de lâmina com ruído. O artigo apresenta outros objetivos como qual é o tamanho médio das RBCs e como pode contornar os problemas apresentados por diferentes condições das imagens (imagens degradadas

ou ruidosas, diferentes técnicas de coloração de sangue, vários tipos de iluminação do microscópio e sobreposição de células adjacentes). O algoritmo é preciso e robusto para determinar a distribuição de células de sangue. Foi ressaltado que não foram abordados problemas como formas de RBC deformadas (lágrimas, crescentes, agulhas ou uma variedade de outras formas), RBCs infectadas e nem células extras sobrepostas que podem ser encontradas em certos tipos de doenças. O algoritmo possui um método que compreende nove etapas: Aquisição e conversão da imagem para o canal verde; Eliminação de ruído com Bivariate; Shrink Wavelet; Preservação de borda com filtro Kuwahara; Binarização com Niblack & Otsu; Preenchimento de objetos fechados; Extração de tamanho; Extração de subimagem contendo a região WBC fechada individual; Separação de WBCs de RBCs; e Algoritmo de contagem. O artigo detalha cada etapa e o porquê de cada escolha feita, focando sempre no melhor resultado possível. Os autores destacam que as principais contribuições são: mitigar o problema de desfoque usando o filtro Kuwahara para a preservação das bordas das células; introdução de novas imagens binárias calibradas usando limiares combinados de Otsu e Niblack; e separação de células de sangue em duas subimagens usando um algoritmo baseado em máscara.

Zhang *et al.* (2019) desenvolveram um algoritmo baseado em YOLOv3 para contar RBCs (individuais e aglomeradas) e WBCs (individuais). O algoritmo contém cinco etapas: aquisição da imagem; pré-processamento; segmentação; extração de características; e contagem de células. O processo é iniciado usando YOLOv3 para células sanguíneas individuais e utiliza método de estimativa de densidade de imagem para as células aglomeradas e, no final, combina os resultados. Os autores afirmam terem feito algumas modificações na estrutura do YOLOv3, que usa a ideia do ResNet para construir Darknet-53 como espinhal dorsal. Porém, os autores se inspiraram pela DenseNet, que no final construíram a Densenet-25 (incluindo 25 camadas de convolução) como a espinha dorsal do YOLOv3. Com fluxogramas, imagens e tabelas, é possível observar e comprovar que o método proposto alcançou as expectativas e o objetivo almejados.

Os cinco artigos e mais dois citados da Pesquisa 01 demonstraram diferentes formas e métodos para se identificar e classificar as RBCs e as WBCs, alguns juntamente com super-resolução ou com ML, outros com conjunto de técnicas baseando em formato ou morfologia, podendo até ser híbrido. Assim, todos

apresentaram técnicas interessantes para as imagens que contém RBCs e WBCs, contribuindo para o desenvolvimento da pesquisa.

2.2.6 Técnicas para Imagens das Principais Células Sanguíneas

Como mostra o Quadro 8, os artigos apresentaram técnicas para reconhecer e categorizar as principais células sanguíneas, que são as RBCs, as WBCs e as plaquetas.

Quadro 8 - Visão Geral dos Artigos da Pesquisa 02 da Sessão Técnicas para Imagens de Todas as Principais Células Sanguíneas

Artigo	Tema/Categoria	Técnicas propostas	Técnicas comparadas	D	A	C
Huang <i>et al.</i> (2016a)	Sangue - Todos	CNNsR	-	X	X	X
Kasamsumran <i>et al.</i> (2020)	Sangue - Todos	MicrosisDCN (Faster R-CNN)	-	X	X	X
Mundhra <i>et al.</i> (2017)	Sangue - Todos	Shonit (U-Net)	-	X	X	X
Khashman (2012)	Sangue - Todos	<i>Morphological Techniques</i>	-	X	X	X
Xia <i>et al.</i> (2020)	Sangue - Todos	YOLOv3	YOLOv3-tiny/YOLOv3-SPP	X	X	X
Deshpande, Gite e Aluvalu (2021)	Sangue - Todos	<i>Review</i>	<i>Decision tree classifier/Random forest/KNN/Logistic Regression/Binary logistic regression model/Ordinal regression/Naïve Bayes algorithm/SVM/CNN</i>	X	X	X
Tessema <i>et al.</i> (2021)	Sangue - Todos	YOLOv2	-	X	X	X
Shakarami <i>et al.</i> (2021)	Sangue - Todos	FED	-	X	X	X
Shahzad <i>et al.</i> (2020)	Sangue - Todos	VGG16	-	X	X	X

*D = Detalhamento das técnicas (Algoritmos)

*A = Aplicações microscópicas para células sanguíneas

*C = Classificação/Identificação de células sanguíneas

Fonte: autoria própria.

Alguns autores usaram também técnicas de super-resolução para melhorar a qualidade das imagens (Huang *et al.*, 2016a), ou ML (Kasamsumran *et al.*, 2020) ou

deep learning (Mundhra *et al.*, 2017) para contar as células. Outros focaram em diferentes modelos de rede neural (Khashman, 2012) ou fundamentando em DNN, como o YOLOv3 (Xia *et al.*, 2020), ou fizeram uma revisão de diferentes técnicas para analisar as células sanguíneas para detecção de doenças com a inteligência artificial (Deshpande, Gite, Aluvalu, 2021). Já Tessema *et al.* (2021) elaboraram um sistema constituído por *hardware* e *software*, usando a arquitetura de YOLOv2, enquanto Shakarami *et al.* (2021) desenvolveram *Fast and Efficient* YOLOv3, ambos para a detecção das células sanguíneas. Shahzad *et al.* (2020), por sua vez, criaram um algoritmo capaz de segmentação semântica em conjunto com VGG16 para reconhecer as RBCs, WBCs e plaquetas. Por fim, Huang *et al.* (2016a), que já foi citado na Pesquisa 01, apresentaram o desenvolvimento de um sistema juntamente com algoritmo de super-resolução que classifica e identifica as células sanguíneas.

Kasamsumran *et al.* (2020) desenvolveram um algoritmo baseado em aprendizado de máquina com processamento múltiplo de GPU, sendo o *Faster R-CNN* para detecção de hematócitos (células sanguíneas), conhecido como MicrosisDCN (*Microbes Diagnosis with Deep Convolutional Neural Network*) em linguagem Python, sendo separado em três grupos: RBCs, WBCs e plaquetas. O algoritmo é composto por “degraus”, onde mais “degraus” implica em mais características que o algoritmo consegue absorver para depois melhor classificar as células sanguíneas. A precisão média foi monitorada desde 3000 degraus até 86000 degraus, com a precisão média de detecção do primeiro entre 85% e 91% e o segundo entre 97% e 99%, demonstrando que o algoritmo atingiu uma acurácia alta.

Mundhra *et al.* (2017) estruturaram um sistema que faz a análise automática de imagens microscópicas de sangue que usa técnicas de *deep learning* conhecido como Shonit TM. O sistema é composto por uma microscopia automatizada de baixo custo e com *software* embutido. O algoritmo usa *deep learning* (U-Net) com processamento de imagens para localização e classificação de células sanguíneas, sendo que os três tipos têm formato redondo e são diferenciados apenas pelo tamanho. Além disso, tem a vantagem de ser capaz de analisar lâminas de sangue preparadas por máquinas ou manualmente. A análise é realizada em etapas separadas, sendo a primeira a extração na qual as células dos três tipos principais são localizadas separadamente. Em seguida, a etapa de classificação, onde são categorizadas em subtipos. Além disso, tem uma classe considerada como fragmentada ou “inválida” que é usada para rejeitar artefatos, células aglomeradas ou

sobrepostas, células degeneradas etc. O artigo apresenta as tabelas e imagens que comprovam que o equipamento proposto possui alta precisão e atingiu os resultados esperados.

Khashman (2012) elaborou diferentes modelos de rede neural com o objetivo de identificação de células (RBCs, WBCs e plaquetas). As células são diferenciadas pelos tamanhos e características morfológicas (possuir ou não núcleo, formato), além de área, cor do núcleo, compacidade, entre outros. O processo dos algoritmos de identificação automatizada possui alguns passos, sendo eles: aquisição, detecção, extração da característica e classificação. O autor utiliza três modelos de rede neural supervisionados baseado em *back propagation*, pois a implementação é simples e a disponibilidade de banco de dados suficiente para treinamento e teste. Após tais testes, a ANN3 foi selecionada, pois apresentou o melhor desempenho devido ao seu tempo de execução mais curto e sua consistência na identificação. O desempenho alto se deve ao treinamento das redes usando vetores de características aproximadas que foram extraídas a partir das imagens de células, com uma taxa geral de identificação correta de 99,17% com 0,0004 s de execução. O algoritmo manteve os custos de tempo no mínimo por meio do pré-processamento e redução de neurônios da camada de entrada e oculta das redes neurais. O autor revela que no estilo em que os dados são apresentados faz uma dependência direta com o modelo do sistema neural, como também seu desempenho. Desse modo, o autor afirma que os dados de entrada precisam ser cautelosamente processados antes de colocar em funcionamento a rede neural focada em características importantes nas imagens e acabar utilizando menos recurso computacional.

Xia *et al.* (2020) criaram um algoritmo baseado em DNN, o YOLOv3, para fazer detecção de contagens completas de células sanguíneas (CBC), que são todas as células sanguíneas principais (RBC, WBC e plaquetas), com o objetivo de facilitar e aumentar a velocidade do diagnóstico e para a implementação em dispositivos microfluídicos em POCs. Além disso, o método proposto usa como caso de estudo as células sanguíneas de sangue de pacientes com COVID-19, nas quais um dos sintomas é a coagulação das células sanguíneas. Os autores fizeram também o uso de outras duas versões de YOLOv3 chamado YOLOv3-tiny e YOLOv3-SPP para os experimentos. Com os resultados, o modelo YOLOv3-SPP demonstra o melhor desempenho entre os três, mesmo que o YOLOv3-tiny tenha um desempenho muito mais rápido, mas apresenta precisão baixa quando aplicado no cenário real. Além

disso, com amostras de pacientes com COVID-19, é possível fazer a identificação de um coágulo de sangue. Os autores finalizam afirmando que os resultados do teste mostraram um perfeito desempenho nessas amostras COVID-19, implicando em um potencial de uma tecnologia para hematologia automatizada para análise médica do COVID-19. Os autores afirmam que existem desafios na detecção como RBCs sobrepostas, imagens ruidosas e com baixo contraste de cores, porém os resultados mostram-se extremamente satisfatórios, principalmente para as amostras de COVID-19 que foi o foco do método proposto.

Deshpande, Gite e Aluvalu (2021) fizeram uma revisão de diferentes técnicas para análise de células sanguíneas para detecção de doenças através da perspectiva da inteligência artificial. O artigo mostra diferentes métodos de classificação e segmentação, focando em explicar o processamento da imagem e quais algoritmos que podem ser usados e quais seriam melhores. O artigo também destaca diversos problemas enfrentados quando são usados algoritmos para identificar e detectar as células, como o problema de células sobrepostas, sendo a revisão destinada a pesquisadores, médicos, patologistas, entre outros que possuem o objetivo de entender e compreender melhor essa área de pesquisa. No artigo, são apresentados fluxogramas que mostram quais são os componentes do sangue e outro que mostra o funcionamento geral de um algoritmo com as etapas de pré-processamento, segmentação, extração de características (cor, geometria, estatística, textura [entropia, contraste, energia], correlação, homogeneidade e direção), classificação (algoritmos como: *Decision tree classifier*, *Random forest*, KNN, *Logistic regression*, *Binary logistic regression model*, *Ordinal regression*, *Naïve Bayes algorithm*, SVM, CNNs, sendo bem detalhado com vantagens e desvantagens) e detecção da doença. O artigo mostra que através da contagem de células é possível identificar qual doença o paciente está sofrendo, como doenças virais (malária, dengue, Chikungunya, hepatite), leucemia, anemia ou célula falciforme. E, para tomar as decisões, diferentes classificadores podem ser projetados. O artigo mostra também as bases que são disponibilizadas para serem usadas nos estudos como *BCCD Database*, *Atlas of hematology by Nivaldo Mediros*, *ASH image bank*, *C-NMC dataset*, entre outros. No final, encontram-se diversas tabelas que comparam as bases disponíveis, comparam diferentes métodos de detecção de doenças de diferentes autores e ainda finaliza com análise de "lacunas" existentes na área. Os autores ressaltam que a inteligência artificial é usada principalmente para detecção e diagnóstico em imagens médicas.

Entretanto, geralmente não é preferido como a decisão final sobre o diagnóstico, pois a principal razão para isso é que funciona como uma caixa preta com apenas entradas e saídas conhecidas.

Tessema *et al.* (2021) desenvolveram um algoritmo responsável por detectar, classificar e analisar morfologicamente e automaticamente células sanguíneas (RBCs, plaquetas, basófilos, eosinófilos, linfócitos, monócitos e neutrófilos) de imagens microscópicas usando aprendizagem profunda, principalmente para ajudar em diagnóstico de doenças relacionadas ao sangue. O sistema é composto por partes de *hardware* e de *software*, sendo que o *software* desenvolvido é feito a partir da arquitetura de YOLOv2 para detecção e classificação. Para a avaliação do sistema, a *performance* foi avaliada por matrizes de confusão que demonstraram a alta precisão, porém os autores afirmam que precisam de mais imagens para aumentar ainda mais a acurácia.

Shakarami *et al.* (2021) elaboraram um algoritmo rápido e eficiente YOLOv3 (FED), com o objetivo de detectar de forma rápida e eficiente as diferentes células sanguíneas (RBCs, WBCs e plaquetas). É considerado um detector de um estágio, pois recebe uma imagem de entrada e aprende as probabilidades da classe e as coordenadas da caixa delimitadora, tratando o objeto de detecção como um problema de regressão simples. Embora esses modelos sejam muito mais rápidos do que os detectores de objeto de dois estágios, alcançam menores resultados de precisão. O algoritmo proposto possui algumas mudanças comparado ao algoritmo YOLOv3 original. Essas mudanças foram a diminuição dos parâmetros do modelo usando *Depthwise Separable Convolution Method*, aumento da visão receptiva do detector utilizando *Dilated Convolution*, aumento da eficiência e da flexibilidade do detector usando EfficientNet CNN. Seu método de escalonamento é composto e dispõe da capacidade de treinar e executar o modelo com uma GPU e CPU *midcore*. Isso permite a possibilidade de usabilidade do modelo em sistemas embarcados e equipamentos portáteis, entre outras mudanças. As tabelas demonstram os resultados obtidos com a precisão média de 90,25% para plaquetas, 80,41% para RBC e 98,92% para WBC. Além disso, mostram que houve uma redução no número de parâmetros usados (em relação a outros algoritmos). Apresenta, também, melhora na porcentagem de precisão média por mudanças nas etapas dentro do algoritmo, ressaltando que 80% dos dados foram usados para treinamento e os 20% restantes para a avaliação.

Shahzad *et al.* (2020) estruturaram um algoritmo capaz de segmentação semântica (capaz de diferenciar uma imagem em diferentes porções ou objetos) em conjunto com VGG16 para RBCs, WBCs e plaquetas. A técnica de segmentação consiste em cinco etapas. Primeiro, as imagens originais, juntamente com as geradas por máscaras, são carregadas no sistema para pré-processamento. Na próxima etapa, durante o pré-processamento, são utilizadas máscaras para diferenciar e classificar as células sanguíneas que usam uma proporção de *pixels* em cada célula para aumentar a acurácia e a contagem. Em seguida, cada imagem é redimensionada de acordo com a entrada padrão. A partir disso, é feita a etapa de codificação-decodificação convolucional profunda, na qual a arquitetura VGG16 é encarregada com o modelo pré-treinado para extração de recurso. E, por fim, as imagens de células são divididas em treinamento e teste. Para a avaliação do algoritmo, são utilizados três tipos de matrizes de avaliação: pontuação, IoU e BF e foi usado o banco de dados de imagens de leucemia linfoblástica aguda (TODOS-IDB1). O algoritmo mostrou-se muito útil para o diagnóstico de doenças relacionadas às células do sangue e contagem de células, demonstrando que fornece o número exato de contagem de células sanguíneas, com tempo menor e com alta precisão.

Os nove artigos apresentados demonstraram diversos métodos que utilizam algoritmos para identificar e classificar as principais células sanguíneas. Desde algoritmos envolvendo ML até *deep learning*, todos com seus detalhes e passos adicionados ou modificados para melhorar a técnica. No geral, todos divulgaram métodos úteis e proveitosos para as imagens de RBCs, WBCs e plaquetas, colaborando para esta pesquisa.

Quadro 9 - Destaque da Pesquisa 02

Deshpande, Gite e Aluvalu (2021)	Sangue - Todos	Review	Decision tree classifier/Random forest/KNN/Logistic Regression/Binary logistic regression model/Ordinal regression/Naïve Bayes algorithm/SVM/CNN	X	X	X
----------------------------------	----------------	--------	--	---	---	---

Fonte: autoria própria.

Assim, após toda a sessão da Pesquisa 02, tem-se um destaque especial para o último tópico, mais especificamente para um trabalho mais semelhante à proposta desta pesquisa, do autor Deshpande, Gite e Aluvalu (2021), mostrado no Quadro 9. Os autores fizeram uma revisão de diferentes técnicas para análise de células sanguíneas para detecção de doenças através da perspectiva da inteligência artificial. O artigo mostra diferentes métodos de classificação e segmentação, focando em explicar o processamento da imagem e quais algoritmos que podem ser usados e quais seriam melhores. Entretanto, não existe um trabalho em que houvesse uma revisão e comparação de técnicas de super-resolução juntamente com os métodos de identificação e classificação, tornando-se um dos diferenciais deste trabalho.

3 MATERIAIS E MÉTODOS

3.1 MATERIAIS

Como a pesquisa propõe o desenvolvimento de um algoritmo, os únicos materiais necessários foram um computador que tivesse processamento suficiente para rodar o programa e uma base de dados de imagens que fosse possível utilizar para treinamento e teste.

3.2 CLASSIFICAÇÃO E DELINEAMENTO DO ESTUDO

O estudo é considerado transversal, pois o fator e efeito são observados num mesmo momento histórico e experimental, haja visto que o objetivo é testar uma hipótese.

3.3 METODOLOGIA

Para a fundamentação teórica, foram feitas duas pesquisas, sendo usada a metodologia PRISMA. A Pesquisa 01 teve o intuito de abordar as técnicas de super-resolução em geral que existem na área da saúde.

Na base de dados Web of Science foi usado o seguinte *query*: "(("Super Resolu*") AND ("Sangue" OR "Hemo*" OR "Blood*"))" conseguindo um total de 224 artigos. Com um documento excluído por ser de outra língua estrangeira. Além disso, não houve nenhuma duplicata. Após a leitura do título e do *abstract* de 223 artigos, foram retirados da pesquisa 191 por estarem fora do escopo. Logo, restando um total de 32 artigos que estão dentro ou tangentes ao escopo, como mostrado no Quadro 10. É possível observar os passos da Pesquisa 01 no fluxograma PRISMA na Figura 04.

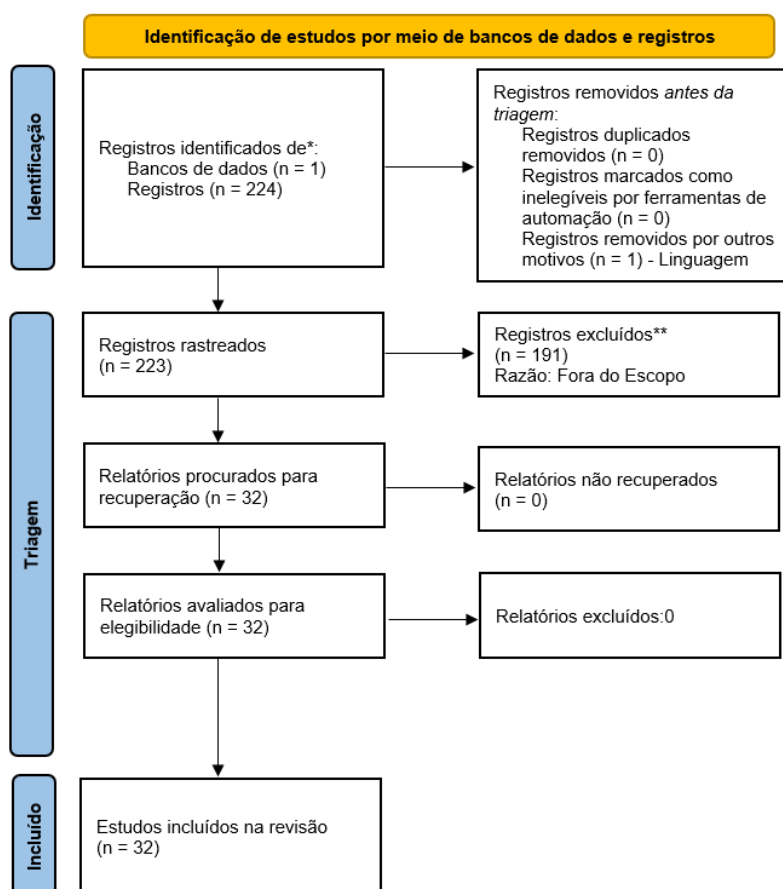
Após a leitura na íntegra dos 32 artigos, foram feitas resenhas e análises de cada um, agrupando por técnicas semelhantes e extraíndo as principais características. Entretanto, percebeu-se por ser uma pesquisa inicial, uma “lacuna” no quesito algoritmos e códigos que abordassem a parte de melhoramento de resolução de imagens. Foi necessária a elaboração de uma segunda pesquisa que, além de abordar os códigos e algoritmos, foi utilizada para pesquisar também sobre identificação e classificação das células sanguíneas, que também faz parte desta pesquisa.

Quadro 10 - Categorização dos Artigos da Pesquisa 01

Pesquisa 01	
Critérios de exclusão:	
Duplicata - D	0
Não há acesso - A	0
Idioma - I	1
Fora do escopo - F	191
Tangente do escopo - T	27
Dentro do escopo - E	5
Total:	32

Fonte: autoria própria.

Figura 4 - Diagrama PRISMA da pesquisa 01



Fonte: autoria própria.

Para a Pesquisa 02, como foi citado anteriormente, o objetivo foi abordar mais sobre algoritmos e códigos para as técnicas de super-resolução de imagens e para a identificação e a classificação de células sanguíneas. A pesquisa foi feita em três

bases de dados: Web of Science, Scopus e *ACM-Digital Library*. Foi utilizada a seguinte *query*: “(("Super Resolu*" OR "Super-Resolu*" OR "Low Resolution" OR "LR" OR "High Resolution" OR "HR" OR "Microscop* Imag*") AND ("Blood* Cell*" OR "Red Blood Cell*" OR "RBC*" OR "White Blood Cell*" OR "WBC*" OR "Plaquet*") AND ("Algorithm*" OR "Code*" OR "Artificial Intelligence" OR "Machine Learning" OR "Deep Learning" OR "Convolutional Neural Network*" OR "CNN*" OR "Generative Adversarial Net*" OR "GAN*") AND ("Count*" OR "Identif*" OR "Detect*" OR "Classif*" OR "Quantif*"))”. Com isso, foram obtidos um total de 699 artigos (Web of Science = 254, Scopus = 394 e ACM = 51). Como a pesquisa foi feita em três bases, sendo duas delas as maiores, ocorreram duplicatas que somaram 221 artigos retirados (Scopus = 219, ACM = 2), atingindo 478 artigos para analisar o título e o abstract. Depois de avaliados, 374 artigos foram removidos por estarem fora do escopo. Resumindo, um total de 104 artigos foram separados para serem lidos na íntegra. Entretanto, um artigo foi retirado por ser de outra língua estrangeira e 20 por não possuírem acesso aberto disponível, resultando em 83 artigos lidos por estarem dentro ou tangente ao escopo da pesquisa (Web of Science = 65, Scopus = 13 e ACM = 3). Contudo, dois artigos foram retirados após serem lidos, pois não apresentaram o escopo correto. Os resultados da pesquisa estão resumidos no Quadro 11.

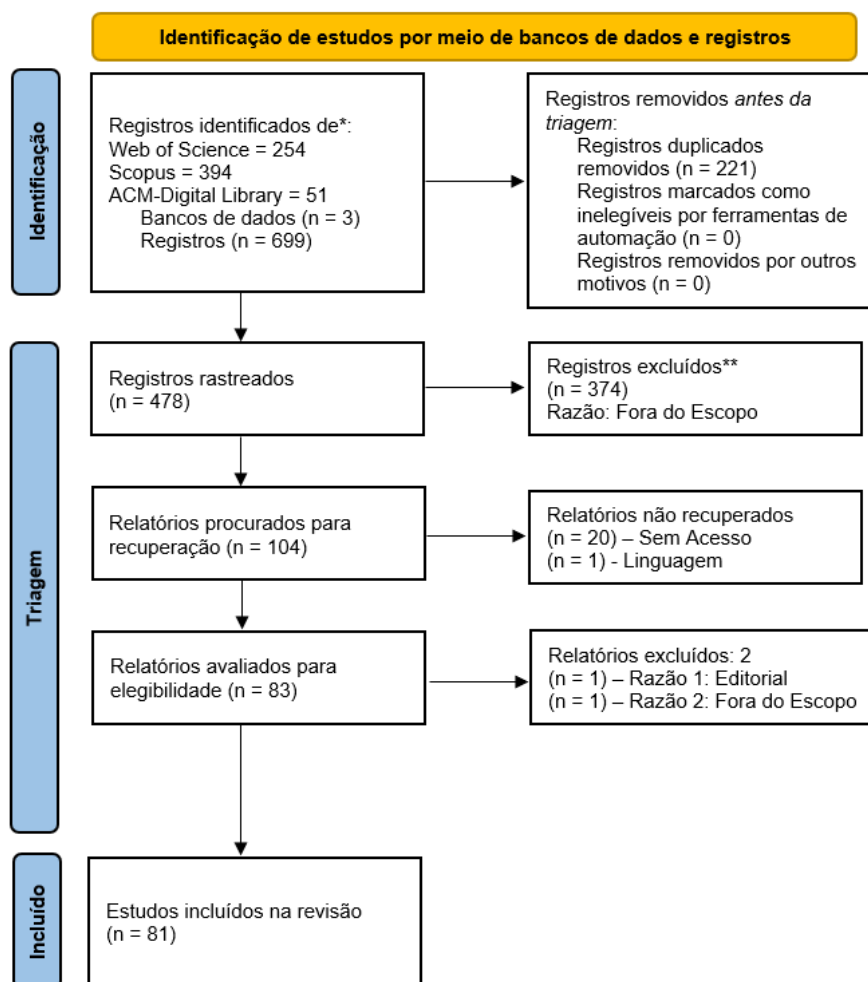
Quadro 11 - Categorização dos Artigos da Pesquisa 02

Pesquisa 02 - Web		Pesquisa 02 - Scopus		Pesquisa 02 - ACM	
Critérios de exclusão:		Critérios de exclusão:		Critérios de exclusão:	
Duplicata - D	0	Duplicata - D	219	Duplicata - D	2
Não há acesso - A	5	Não há acesso - A	13	Não há acesso - A	2
Idioma - I	0	Idioma - I	1	Idioma - I	0
Fora do escopo - F	184	Fora do escopo - F	148	Fora do escopo - F	44
Tangente do escopo - T	59	Tangente do escopo - T	13	Tangente do escopo - T	3
Dentro do escopo - E	6	Dentro do escopo - E	0	Dentro do escopo - E	0
Total:	65	Total:	13	Total:	3
		Total Pesquisa 02:	81		

Fonte: autoria própria.

É possível observar as etapas da Pesquisa 02 no fluxograma PRISMA na Figura 05. Com isso, após concluir essas pesquisas, os objetivos específicos 1 e 2 foram atingidos.

Figura 5 - Diagrama PRISMA da Pesquisa 02



Fonte: autoria própria.

3.4 MÉTODO PROPOSTO

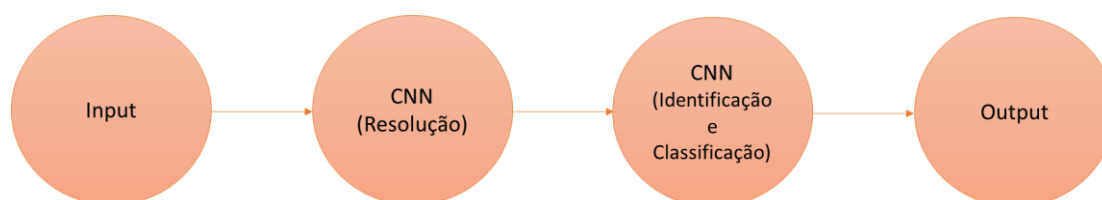
O método proposto foi o desenvolvimento de um algoritmo que consiga melhorar a resolução de imagens microscópicas de células de sangue e que também seja possível classificar as diferentes células sanguíneas (RBCs, WBCs e plaquetas), com o objetivo de observar se ocorre a melhora das métricas tanto de super-resolução e de classificação.

Os algoritmos foram baseados em CNN usando as diferentes camadas para melhorar a resolução das imagens e outra parte somente para extrair as características de cada célula sanguínea e classificá-las, como mostra a Figura 6.

Com base nos resultados anteriores, foi possível encontrar as melhores técnicas, para alcançar os melhores resultados. Com isso, foi implementado duas CNNs, uma para a melhorar a resolução e outra para identificação e classificação, que juntando o resultado final será uma imagem com uma melhora de resolução e com as células identificadas e classificadas. Serão feitos testes para observar qual é o melhor tipo de CNN que será utilizada, sendo que existem diversos modelos e alguns poderão ser escolhidos para serem implementados e alcançarem o resultado mais próximo do ideal.

Para avaliar o desempenho dos algoritmos que foram implementados, foram usadas as quatro métricas mais comuns, citadas anteriormente, sendo em alinhamento com os estudos recentes, que atingiram os valores superiores a 85%.

Figura 6 - Esquema Geral do Método Proposto



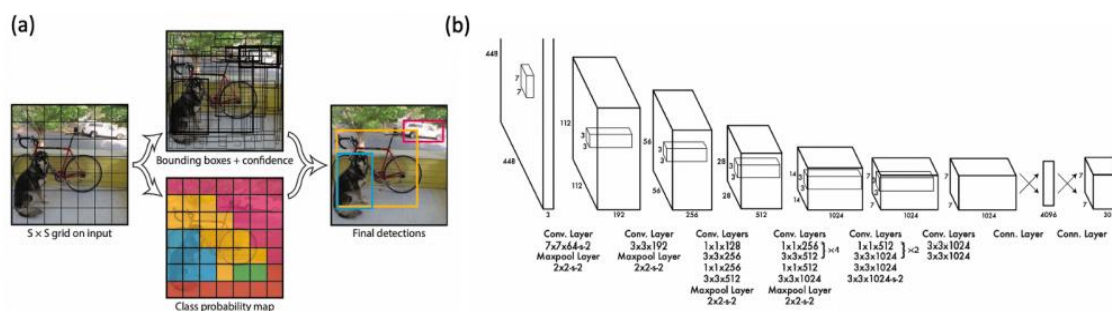
Fonte: autoria própria.

A CNN foi escolhida por apresentar com maior destaque na literatura e por divulgar ótimos resultados tanto em precisão quanto em tempo de processamento, como foi mostrado na seção 2 “Fundamentação Teórica”. Assim, diferentes arquiteturas CNN foram testadas para observar qual apresenta melhores resultados, como por exemplo, YOLO, *Inception*, VGG16 (que foram citadas anteriormente), entre outras.

A YOLO, de acordo com Redmon *et al.* (2016) e a revisão de Oliveira *et al.* (2021), é uma rede unificada extremamente rápida para localização e classificação de objetos, aprendendo diversas representações gerais dos objetos pedidos, com base em somente um *backbone* de CNN que utiliza regressão, que usa caixas delimitadoras que localizam o alvo requerido juntamente com um rótulo mostrando a probabilidade. A arquitetura é demonstrada na ilustração da Figura 07. Como é possível observar, a

Figura 7-A mostra o funcionamento do algoritmo, localizando e marcando os objetos desejados e a Figura 7-B ilustra o fluxo das camadas convolucionais que se encontram dentro da arquitetura.

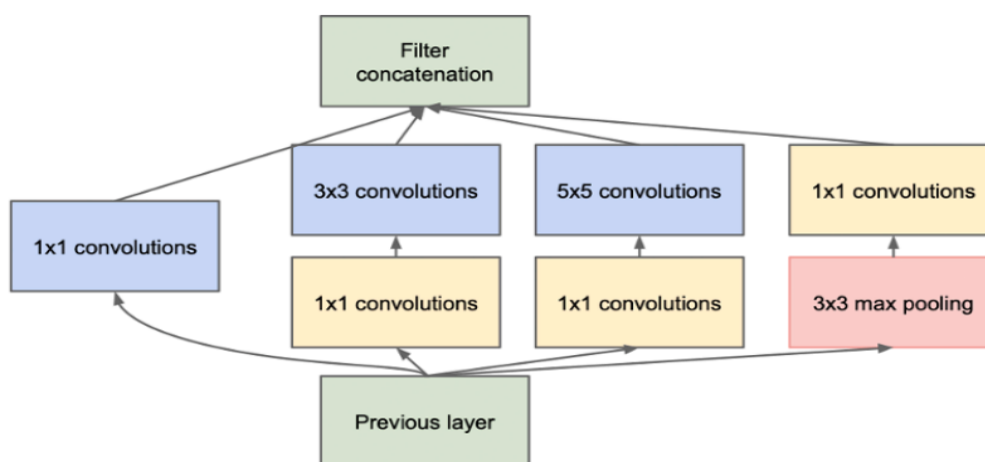
Figura 7 - Algoritmo YOLO



Fonte: Redmon *et al.* (2016).

O algoritmo *Inception*, proposto por Szegedy et al. (2015) e a revisão de Oliveira et al. (2021), é utilizado para extrair de forma numerosa as características de objetos locais e treina diferentes camadas convolucionais simultaneamente e organiza as redes com as características adquiridas, classificando a imagem como um todo. A Figura 8 demonstra um exemplo de um módulo com base no algoritmo *Inception* com dimensões reduzidas.

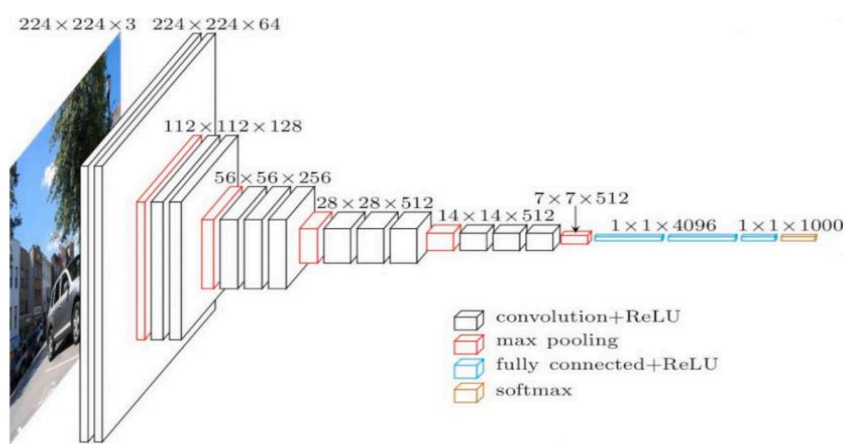
Figura 8 - Exemplo de Módulo de Inception com dimensões reduzidas



Fonte: Szegedy *et al.* (2015).

Por fim, um último exemplo de arquitetura que foi testada é o modelo VGG16, que de acordo com Simonyan e Zisserman (2014) e a revisão de Oliveira *et al.* (2021), que é principalmente utilizado para a classificação de imagens, composto por dezesseis camadas convolucionais, múltiplas camadas *max-pool* e três camadas finais totalmente conectadas. A Figura 9 mostra como ocorre o processo de funcionamento interno da arquitetura VGG16.

Figura 9 - Algoritmo VGG16

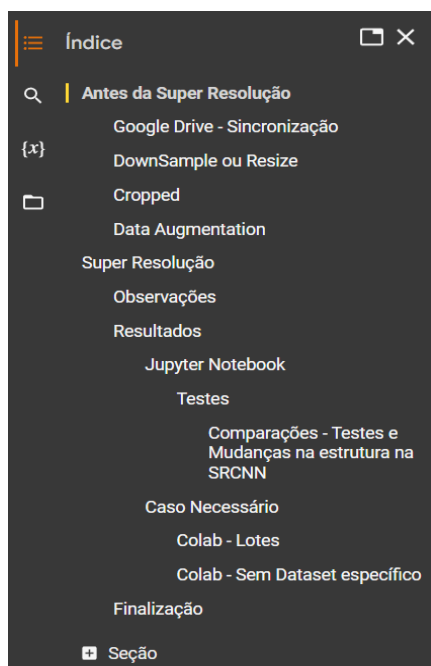


Fonte: Simonyan e Zisserman (2014).

Por fim, esses diferentes tipos de arquiteturas foram ser avaliados pelas métricas citadas na seção 1.4 Hipótese, que são precisão, acurácia, sensibilidade e especificidade, procurando alcançar em todas o valor de 85% ou superior. Desse modo, buscou-se o melhor algoritmo com os parâmetros que se encaixam da melhor forma, focando atingir os resultados mais próximos aos propostos.

Assim, com a CNN sendo a escolha final da estrutura do código, considera-se concluído o objetivo específico 3 e foi iniciado o desenvolvimento do programa, que foi dividido nas seguintes partes (índice) dentro do Google Colab, como mostrado na Figura 10.

Figura 10 - Divisão do Código no Google Colab



Fonte: autoria própria.

O tópico “Antes da Super Resolução” insere todos os passos necessários antes que ocorra o processo da Super Resolução. Dentro desse tópico existem subtópicos como: “*Google Drive - Sincronização*”, “*DownSample ou Resize*”, “*Cropped*” e “*Data Augmentation*”.

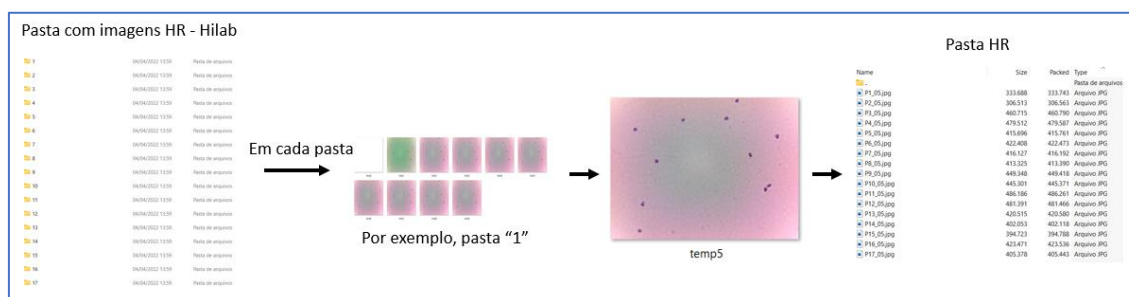
Já no tópico “Super Resolução”, está o detalhamento de cada etapa dos resultados alcançados explicados com seus devidos comentários, esclarecendo o porquê de cada escolha e com gráficos para ter uma melhor observação visual dos valores atingidos. Nesse tópico, existem os subtópicos de “Observações” e “Resultados” que, no último, possui “*Jupyter Notebook*” que dispõe de “Testes”, o qual, por sua vez, detém o subtópico “Comparações - Testes e Mudanças na estrutura da SRCNN”. Novamente, dentro de “Resultados”, existe um outro subtópico chamado de “Caso Necessário” que possui dentro “Colab - Lotes” e “Colab - Sem *dataset* específico”. E por fim, existe o tópico “Finalização”.

O primeiro passo é “Google Drive - Sincronização”, que existe o código responsável por sincronizar o Google Drive com o notebook usado no Google Colab, pois é no Google Drive que foram colocadas as imagens (que serão explicadas no futuro) utilizadas no algoritmo de Super Resolução.

O segundo passo é “*DownSample* ou *Resize*”, com o intuito de diminuir o tamanho das imagens para criar o banco de dados de imagens de baixa resolução (LR). As imagens (amostras de lâminas) foram fornecidas pela empresa Hilab, as quais eram WBCs em alta resolução (HR), que enviou 17 pastas com 10 imagens cada (sendo 10 imagens “iguais” e somente diferentes capturas, podendo mudar a cor e o foco). No caso da pasta ter imagens “iguais”, foi escolhida a quinta imagem, por sugestão da própria empresa e por ter a maior possibilidade de apresentar o melhor foco, alcançando o total de 17 imagens com HR de lâminas de sangue com somente WBCs. Na Figura 11, é possível observar o passo a passo feito, iniciando com as 17 pastas com 10 imagens em cada, escolhendo a quinta imagem e separando para a pasta HR, finalizando com o total de 17 imagens. O fluxograma está na Figura 12.

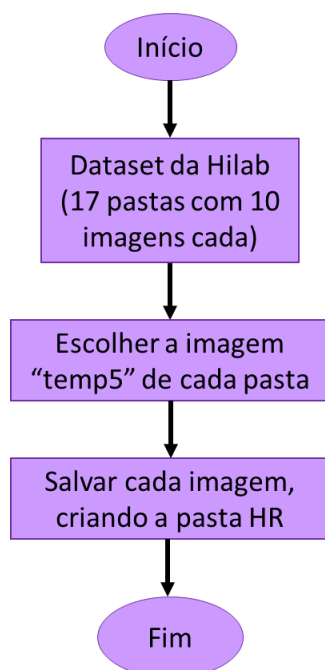
Continuando o segundo passo, com essa pasta que continha as 17 imagens em HR, as mesmas foram “chamadas” dentro do código que faz a parte de redução das imagens para criar o *dataset* de LR, sendo reduzidas em 4x do tamanho original. A empresa Hilab utilizou lentes com 40x de aumento, logo as imagens LR ficaram 10x (de aumento) e necessitam de um algoritmo (de Super Resolução) que faça o melhoramento em 4x para voltar ao mesmo nível que estava antes. Assim, como mostrado na Figura 13, a partir da pasta HR, que detém todas as imagens com 2.592 *pixels* por 1.944 *pixels*, passando pelo código foram reduzidas 4 vezes para 648 *pixels* por 486 *pixels*, sendo agrupadas para fazer o *dataset* de LR com 17 imagens e cujo fluxograma está na Figura 14.

Figura 11 - Separação das imagens



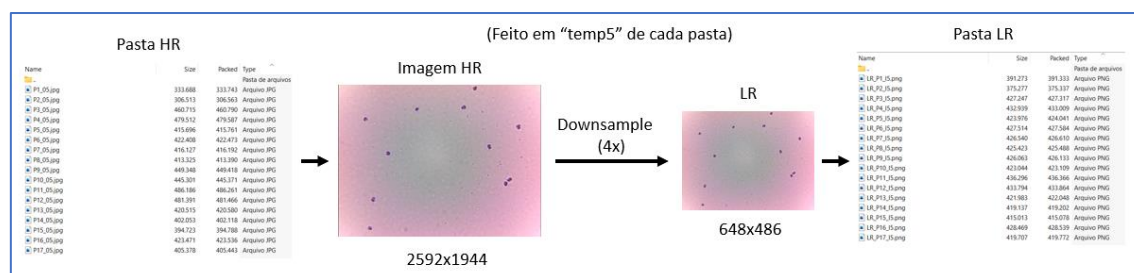
Fonte: autoria própria.

Figura 12 - Separação das imagens - Fluxograma



Fonte: autoria própria.

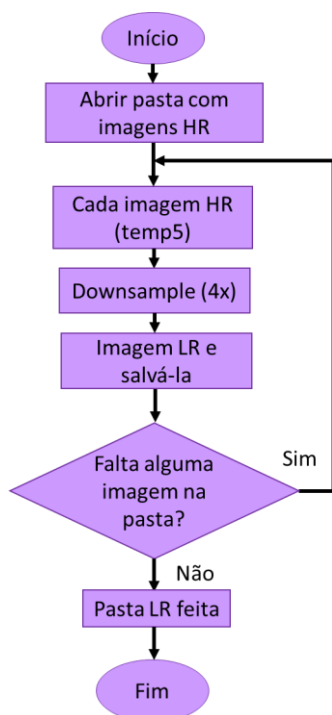
Figura 13 - DownSample ou Resize



Fonte: autoria própria.

Para fazer o *DownSample*, foram feitos dois tipos de códigos. O primeiro, que pode ser usado para somente uma figura, e o segundo, que pode ser usado em uma pasta completa, reforçando que foi usado o segundo método por maior facilidade e que as imagens foram salvas em extensão PNG para que não ocorresse perda na qualidade das imagens.

Figura 14 - DownSample ou Resize - Fluxograma



Fonte: autoria própria.

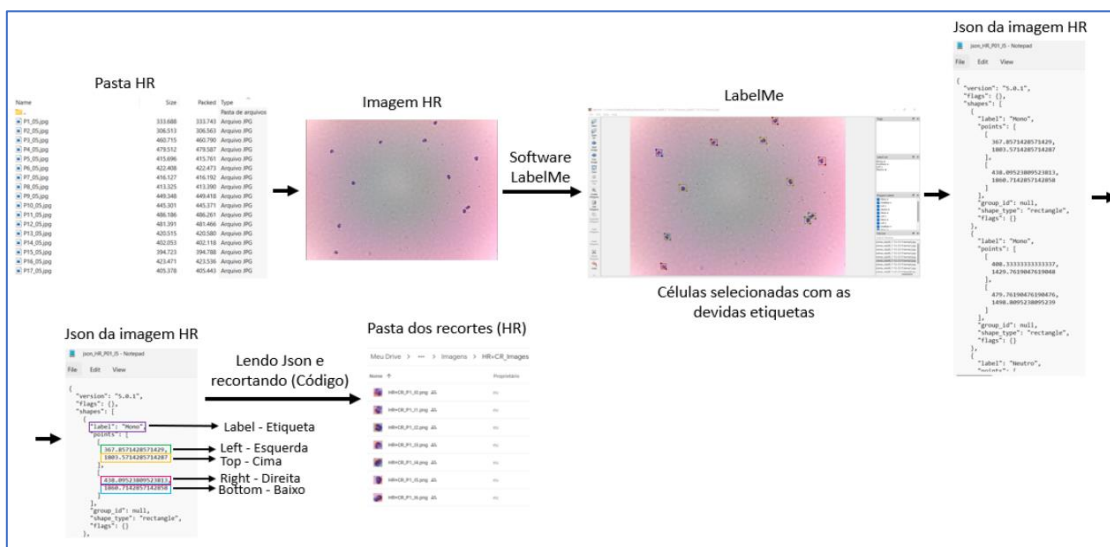
A terceira etapa é “*Cropped*”, com o objetivo de fazer os recortes das células que estão dentro das imagens de HR e LR. Esse recorte extrai cada célula que esteja apta para ser utilizada no código de super-resolução. Foram feitos dois modos para a extração, sendo que o primeiro modo é recortar cada célula individualmente e salvá-la. Entretanto, seria preciso saber os *pixels* certos o que tornaria o trabalho árduo e complicado. O segundo modo (o qual foi utilizado por maior facilidade) é usando o *software* LabelMe (recomendado pela empresa Hilab), o qual é possível selecionar todas as WBCs aptas e também colocar as etiquetas (ou *labels*, que são os tipos das WBCs: neutrófilos, eosinófilos/basófilos, monócitos e linfócitos), sendo que a classificação de cada tipo de célula foi feita pela autora com base no artigo de Gasparin *et al.* (2022) desenvolvido pela própria empresa Hilab. Assim, foram selecionadas as imagens que somente apresentam células em que é possível ver o núcleo, o qual é uma mancha vermelha central. As imagens também foram conferidas por uma pessoa responsável pela verificação da empresa Hilab (se a imagem estava apta e se a classificação estava correta).

Logo após, o *software* LabelMe cria um arquivo .json, salvando todos os recortes feitos dentro da imagem, os quais armazenam dois pontos de cada imagem recortada (topo e esquerda, abaixo e direita) juntamente com a etiqueta.

Na terceira etapa, a empresa retornou a verificação das imagens e das etiquetas (ou rotulações) informando que as rotulações foram corrigidas. Comunicou que a quantidade em porcentagem de cada célula que existe em uma pessoa saudável, sendo elas: neutrófilos entre 40% a 70%, linfócitos entre 15% a 60%, monócitos entre 0% a 12% e eosinófilos juntamente com os basófilos entre 0% a 6%. A pessoa responsável pela verificação também ressaltou que as últimas células são consideravelmente maiores do que as outras células.

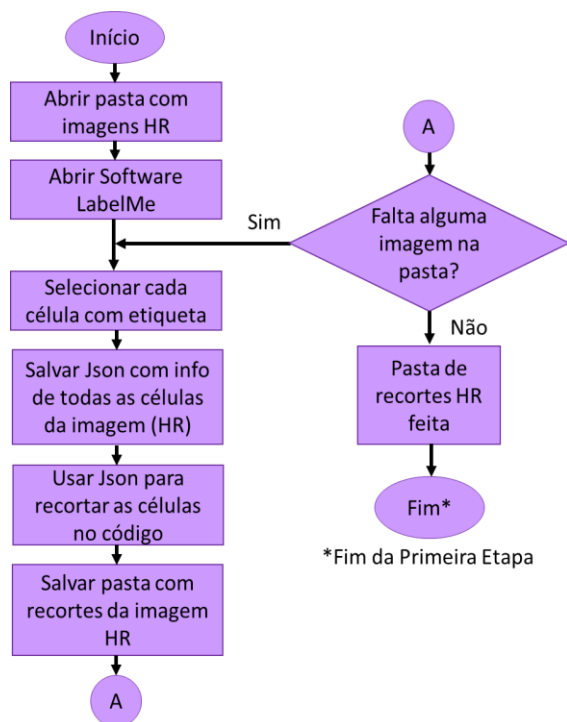
O processo do “*Cropped*” inicia com a pasta de HR, cada imagem dentro da pasta é colocada no *software* LabelMe e cada célula apta é recortada e etiquetada, salvando seus pontos e sua categoria em arquivo json, o qual é lido no código e recorta cada célula salvando em uma nova pasta chamada “Pasta dos recortes (HR)”, como é mostrado na Figura 15 e o fluxograma na Figura 16.

Figura 15 - Cropped - Processo 1



Fonte: autoria própria.

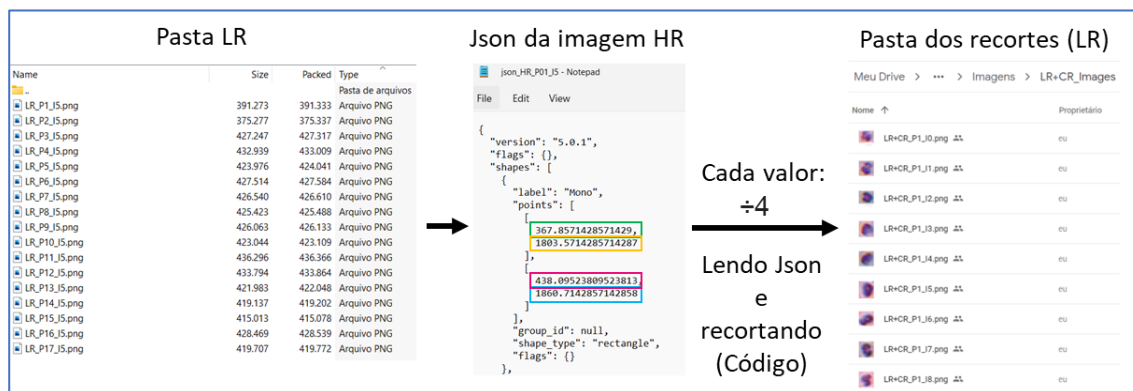
Figura 16 - Cropped - Fluxograma 1



Fonte: autoria própria.

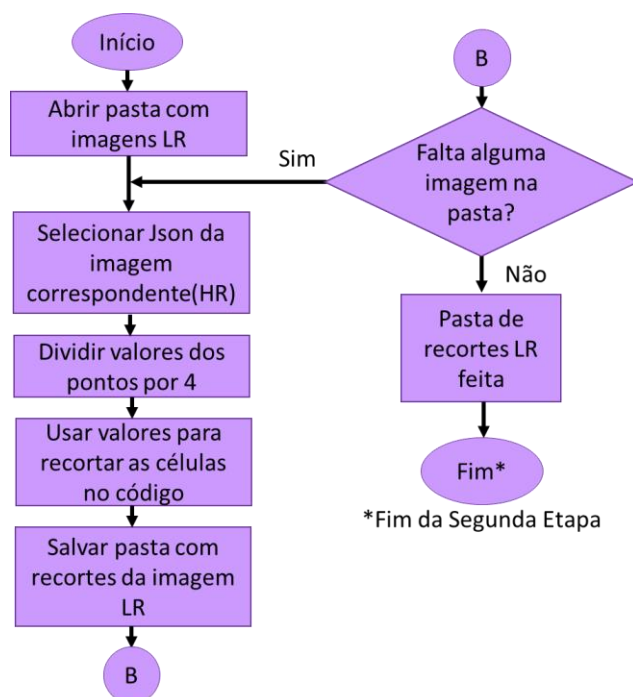
Porém, é necessário fazer o mesmo procedimento para as imagens de LR. Como é observado na Figura 17, é usado o mesmo arquivo json das imagens de HR. Mas, como as imagens são exatamente quatro vezes menor, foi somente necessário dividir os valores dos *pixels* (pontos) por quatro (mantendo a precisão dos recortes com as mesmas proporções entre as imagens de HR e LR) e salvando as novas imagens em uma nova pasta chamada “Pasta dos recortes (LR)”. O fluxograma está apresentado na Figura 18.

Figura 17 - Cropped - Processo 2



Fonte: autoria própria.

Figura 18 - Cropped - Fluxograma 2



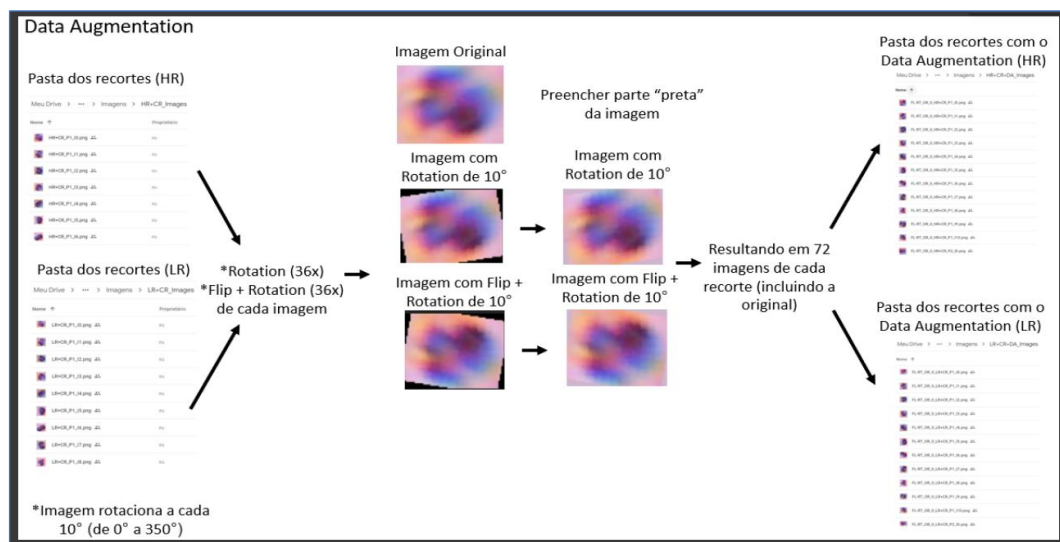
Fonte: autoria própria.

O quarto passo é a “*Data Augmentation*” para o código de SR, que tem como intuito aumentar a quantidade de imagens existentes através de funções que podem desde rotacionar a imagem até mudar a coloração, criando novas figuras a partir da original. Essa técnica é utilizada quando a quantidade de imagens para o treinamento é pouca e é necessário gerar novas para que não ocorra de a rede neural aprender somente as imagens que observou (*Overfitting*), desenvolvendo uma dificuldade em entender outras imagens fora da pasta de treinamento. Para que o modelo seja melhor treinado, é necessário que ele observe sempre imagens diferentes, expondo a mais características dos dados, criando uma generalização maior.

Para a geração de novas imagens (já recortadas) foram usados dois tipos de funções: *Rotation* e *Flipping* com *Rotation*. Para a “*Rotation*”, foram feitas 36 mudanças de 10 em 10 graus (salvando a imagem original no grau 0 e 35 novas imagens com ângulos diferentes até alcançar 350 graus). Já para o “*Flipping* com *Rotation*”, foram usadas as mesmas imagens feitas em “*Rotation*” e aplicada a função *Flip* para virar a imagem, criando mais 36 imagens. No total, foram alcançadas 71 imagens novas a partir da original, sendo tomados todos os cuidados para não criar imagens iguais. Além disso, também foi preenchido o fundo das imagens, pois quando a figura é rotacionada, o seu fundo é preenchido com preto. Para permanecer com a

borda semelhante da imagem original (para manter a uniformidade nas bordas e não prejudicar o aprendizado na rede neural), foi usada uma função para “recriar” o fundo colorido através dos valores dos *pixels* em volta e logo após, salvar em suas devidas pastas. A explicação está na Figura 19 e todo o processo é mostrado na Figura 20.

Figura 19 - Data Augmentation - Processo



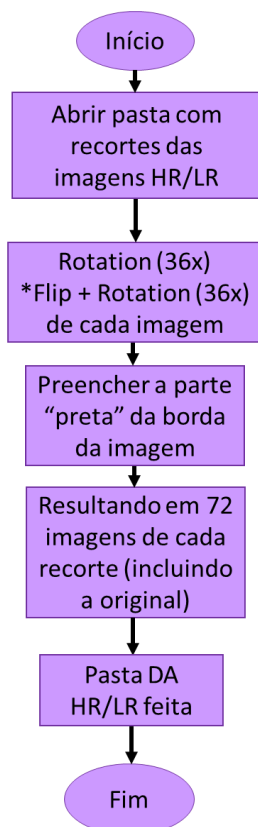
Fonte: autoria própria.

Neste momento, continuando a quarta etapa, em resumo, existiam no total 249 recortes feitos (em cada pasta HR e LR, criados na etapa do “*Cropped*”) e fazendo o cálculo, 249×72 imagens (juntamente com a original), resultará em 17.928 imagens em cada pasta (HR e LR). Assim, como existem as pastas HR e LR, é 17.928×2 que resultará em 35.856 imagens no total, tanto em HR e LR.

Adicionalmente, codificou-se um programa para contar a quantidade de cada tipo de WBCs existente, para seguir e comparar os valores, para ser possível fazer a divisão mais aproximada entre as pastas de treinamento, validação e teste. Das 249 imagens, 195 serão de neutrófilos, 17 serão monócitos, 33 serão linfócitos e 4 serão eosinófilos/basófilos.

Assim, as 249 imagens representarão 100%, sendo que os neutrófilos serão 78%, monócitos serão 7%, linfócitos serão 13% e eosinófilos/basófilos serão 2%, estando um pouco fora do esperado das proporções que a pessoa responsável informou. Entretanto, tais resultados não serão tão diferentes a ponto de serem desproporcionais ao normal.

Figura 20 - Data Augmentation - Fluxograma



Fonte: autoria própria.

Na pasta de treinamento, ficou separado 60% de cada tipo de célula, resultando em um total de 10.728. Já nas pastas de validação e de teste foram divididos aproximadamente 20% das imagens em cada. Como não é possível dividir as imagens na metade, a pasta de validação ficaram com imagens a mais em comparação com a pasta de teste. Na Figura 21, é possível observar a quantidade de imagens em cada pasta, a pasta de validação com 3.672 imagens e a pasta de teste com 3.528 imagens.

Assim, essa foi a separação do *dataset* para somente o código de SR.

Figura 21 - Data Augmentation - Divisão

Treinamento - 60%	Validação - 20%	Teste - 20%
Neutro - $195 \times 60\% = 117 \times 72 = 8424$	Neutro - $195 \times 20\% = 39 \times 72 = 2808$	Neutro - $195 \times 20\% = 39 \times 72 = 2808$
Mono - $17 \times 60\% = 10 \times 72 = 720$	Mono - $17 \times 20\% = 4 \times 72 = 288$	Mono - $17 \times 20\% = 3 \times 72 = 216$
Linf - $33 \times 60\% = 20 \times 72 = 1440$	Linf - $33 \times 20\% = 7 \times 72 = 504$	Linf - $33 \times 20\% = 6 \times 72 = 432$
Eosi - $4 \times 60\% = 2 \times 72 = 144$	Eosi - $4 \times 20\% = 1 \times 72 = 72$	Eosi - $4 \times 20\% = 1 \times 72 = 72$
Tot = $149 \times 72 = 10728$	Tot = $51 \times 72 = 3672$	Tot = $49 \times 72 = 3528$

Fonte: autoria própria.

Após toda a parte de explicações e os códigos do tópico “Antes da Super Resolução”, inicia a parte do tópico “Super Resolução”, começando pelo subtópico “Observações”, a qual comenta que é necessário fazer as imagens com o algoritmo de SRCNN. Entretanto foram feitas diversas tentativas sem sucesso de rodar a rede neural dentro do Google Colab, pois a quantidade de imagens é grande, implicando em um tempo considerável de processamento, fazendo que o Google Colab entrasse no estado “Ocupado” e não terminasse de rodar ou atingir o “Tempo Esgotado” e reiniciasse todo o ambiente, como é mostrado na Figura 22.

Figura 22 - Super Resolução - Tempo Esgotado

```

9 label = np.array(h.get('label'))
10 X_test = np.transpose(data, (0, 2, 3, 1))
11 y_test = np.transpose(label, (0, 2, 3, 1))
12
13 X_train.shape, y_train.shape, X_test.shape, y_test.shape
14 ((52856, 32, 32, 1),
   (52856, 20, 20, 1),
   (185840, 32, 32, 1),
   (185840, 20, 20, 1))
15 [ ]
16
17 1 checkpoint_path = './srcnn/cp-(epoch:844).ckpt'
2 checkpoint_dir = os.path.dirname(checkpoint_path)
3 checkpoint = tf.keras.callbacks.ModelCheckpoint(filepath=checkpoint_dir, save_best_only=True,
4 save_weights_only=True, verbose=0)
5
6 1 srcnn_model.fit(X_train, y_train, batch_size=64, validation_data=(X_test, y_test),
7 callbacks=[checkpoint], shuffle=True, epochs=200, verbose=False)
8
9 Finally, Training is done.
10
11 Predict image from model
12 Let's try it and see its work.

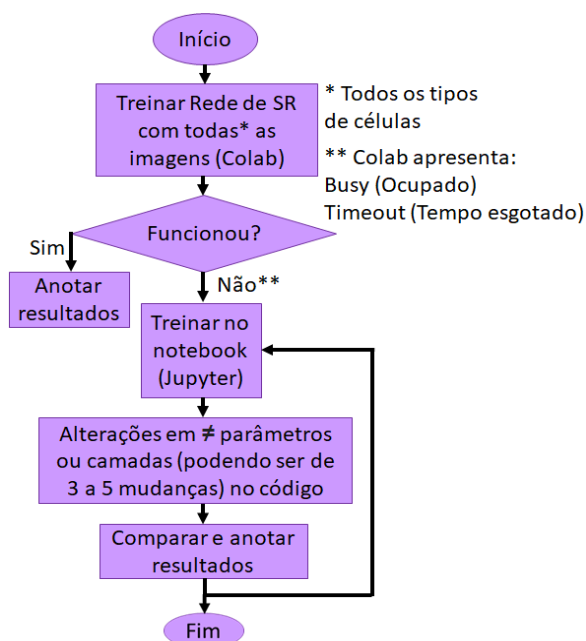
```

Fonte: autoria própria.

Fez-se necessário adotar uma forma diferente de abordagem, que foi treinar o código de Super Resolução no *notebook* da autora, através da plataforma Anaconda com o ambiente *Jupyter*, sendo exatamente o mesmo programa, somente sendo ajustado para rodar e salvar dentro da própria máquina e não mais no Google Colab. Havia a possibilidade de ocorrer de 3 a 5 alterações diferentes dentro do mesmo

código, podendo ser estruturas diferentes (com mais ou menos camadas) ou somente parâmetros diferentes (já existentes dentro das camadas). Assim, com cada mudança, anotou-se e comparou-se o conjunto dos resultados para saber qual estava obtendo melhores valores. A Figura 23 mostra todo o processo descrito anteriormente com um fluxograma detalhado.

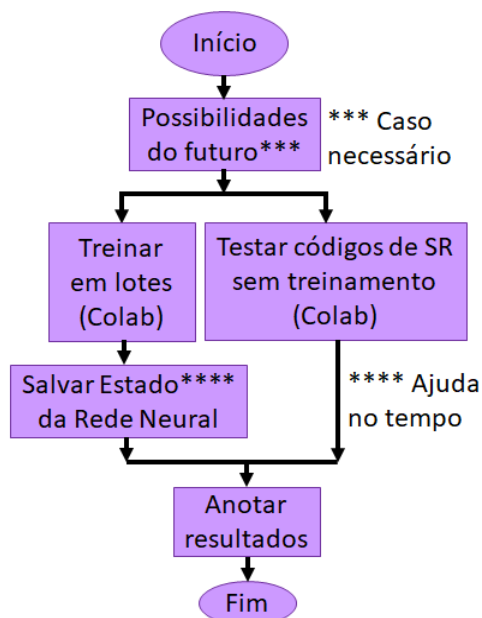
Figura 23 - Super Resolução - Fluxograma Jupyter



Fonte: autoria própria.

Também foram anotadas duas possibilidades para o futuro, caso sejam necessárias: a primeira, que é treinar o programa de Super Resolução no Google Colab, entretanto fazer em lotes para que não ocorra os erros citados anteriormente. Logo também, é preciso salvar o estado da Rede Neural para que consiga prosseguir do estado em que parou. A segunda possibilidade é testar códigos de Super Resolução no Google Colab que não precisam de treinamento com o *dataset* específico, pois eles já possuem um pré-treinamento feito anteriormente com certos "pesos", mostrados na Figura 24.

Figura 24 - Super Resolução - Fluxograma Possibilidades



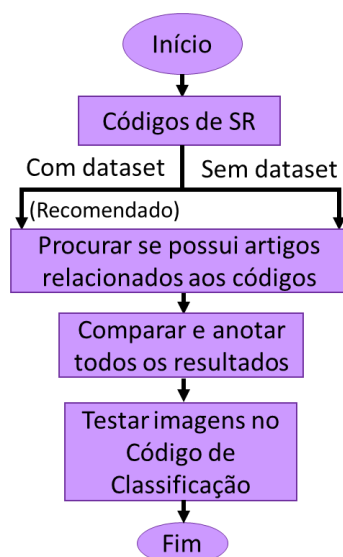
Fonte: autoria própria.

Em qualquer uma dessas abordagens, podendo ser com ou sem *dataset* específicos (se utilizados), foram anotados os resultados e comparados entre eles, além de também ser procurada a existência de artigos que fundamentam os códigos em questão, como mostrado na Figura 25.

Após fazer os testes e comparar os resultados, foi escolhida a melhor estrutura e parâmetros do código e geradas aproximadamente 3000 imagens a partir da pasta “Teste”, criando 3528 imagens artificiais com a Super Resolução. Desse modo, as imagens foram preparadas para serem testadas na CNN de classificação, que também foi feita para observar se houve os resultados esperados escritos na sessão “Hipótese”.

No próximo subtópico, “Resultados”, dentro do índice do Google Colab, foram registrados os resultados alcançados das diferentes abordagens (se necessárias). Relembrando que o código usado para o *Jupyter Notebook* e Colab - Lotes (se for necessário) foram os mesmos, com o objetivo de observar se é possível treinar em lotes sem alterar os resultados alcançados.

Figura 25 - Super Resolução - Fluxograma Finalização



Fonte: autoria própria.

Antes de entrar no próximo tema, é preciso entender quais são as métricas para a avaliação das imagens, que comprovem quantitativamente o melhoramento de qualidade nas imagens. Baseando-se nos artigos de Horé e Ziou (2010); Bansal, Raj e Choudhury (2016); Dong *et al.* (2016); El-Shafai *et al.* (2022) foram usadas as seguintes métricas:

- *Mean Squared Error* (MSE).
- *Peak Signal to Noise Ratio* (PSNR).
- *Structural Similarity Index Measure* (SSIM).
- Laplaciano e Variância do Laplaciano.

A primeira, MSE (em português, Erro Quadrático Médio), definida pelo Site MathWorks, acessado pelo link na data 20/10/2022 (<https://www.mathworks.com/help/vision/ref/psnr.html>), representa o erro quadrático cumulativo entre a imagem compactada e a original. Quanto menor o valor de MSE, menor o erro. A motivação para uso como métrica de qualidade de imagem está em que o erro quadrático médio (MSE) permite comparar os valores de *pixel* “verdadeiros” da imagem original com os da imagem degradada. O MSE representa a média dos quadrados dos “erros” entre a imagem real e a imagem ruidosa. O erro é a quantidade pela qual os valores da imagem original diferem da imagem degradada, a qual é demonstrada na Equação 1.

Equação 1 - MSE

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \cdot \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \quad (1)$$

Fonte: autoria própria.

Assim, na fórmula, I é imagem monocromática e K é sua aproximação ruidosa.

A segunda, PSNR (ou em português, Relação Sinal-Ruído de Pico), também definida pelo Site MathWorks, acessado pelo link na data 20/10/2022 (<https://www.mathworks.com/help/vision/ref/psnr.html>), calcula a relação sinal-ruído de pico, em decibéis, entre duas imagens. Essa proporção é usada como medida de qualidade entre o original e uma imagem compactada. O PSNR é uma métrica objetiva comumente usada para medir a qualidade de reconstrução de uma transformação com perdas, sendo inversamente proporcional ao logaritmo do Erro Quadrado Médio (MSE) entre a imagem real e a imagem gerada. Quanto maior o PSNR, melhor a qualidade da imagem compactada ou reconstruída. A proposta é que quanto maior o PSNR, melhor a imagem degradada reconstruída para corresponder à imagem original e melhor o algoritmo reconstutivo. Isso ocorreria porque deseja-se minimizar o MSE entre imagens em relação ao valor máximo do sinal da imagem, demonstrada na Equação 2.

Equação 2 - PSNR

$$P S N R = 10 * \log_{10} * \left(\frac{MAX_I^2}{MSE} \right) \quad (2)$$

$$P S N R = 20 * \log_{10} * (MAX_I) - 10 * \log_{10} * (MSE) \quad (3)$$

Fonte: autoria própria.

Tanto a MSE e a PSNR são usadas para comparar a qualidade de compressão da imagem.

A terceira, SSIM (ou em português, Medida do Índice de Similaridade Estrutural), também definida pelo site MathWorks, acessado pelo link na data 20/10/2022 (<https://www.mathworks.com/help/images/ref/ssim.html>), é uma função que calcula o índice de similaridade estrutural entre as imagens, usando uma imagem de referência e outra para a comparação. O valor mais próximo de 1 indica melhor qualidade de imagem, que seria mais próximo da imagem original. O SSIM é uma métrica de percepção que quantifica a degradação da qualidade da imagem causada

pelo processamento, como compactação de dados ou por perdas na transmissão de dados. Os valores do SSIM variam entre 0 e 1, onde 1 significa uma correspondência perfeita entre a imagem original e a cópia.

A quarta, e última, é a Laplaciano definida pelo Site OpenGenus, acessado pelo link na data 20/10/2022 (<https://iq.opengenus.org/laplacian-filter/>), e a Variância do Laplaciano, sendo uma métrica de foco de imagem (de *sharpness*). Primeiro é chamada a função do laplaciano na imagem (vetor) e depois é usada a variância, sendo, de fato, um detector de bordas (sendo um filtro passa alta). Nessa métrica, é interpretado que quanto maior o valor, maior o foco que existe na imagem.

Desse modo, ocorrem duas comparações (para as métricas PSNR, MSE e SSIM). A primeira é feita entre a imagem que entra na SRCNN (*input* - que sofre alterações de tamanho e alguns outros processos, como de cor, vetorização e normalização e que por isso, não é uma imagem exatamente igual a se fosse feito o processo de *resized up and down*) e a original (HR). A segunda comparação é feita entre a imagem de saída (*output* - pós SRCNN) e a original, como mostram, respectivamente as Tabelas 1 e 2. E, somente no Laplaciano, é feita a variância do Laplaciano da imagem original e depois da imagem *input* e depois *output*, comparando os dois últimos resultados com a original, como é possível observar na Tabela 3.

Tabela 1 - - Exemplo de Comparação entre Imagem Original e Imagem de Entrada

	Linfocito	Eosi/Basofilo	Monocito	Neutrofilo
PSNR	37.01	34.49	39.32	37.39
MSE	38.85	69.37	22.83	35.61
SSIM	0.97	0.95	0.97	0.96

Fonte: autoria própria.

Após a explicação das métricas, existe dentro de “Resultados”, o subtema “Jupyter Notebook”. Neste foram anotados todos os resultados atingidos por cada mudança de parâmetro no código usado que é SRCNN, ou Super Resolution Convolutional Neural Network. Este usa as redes neurais convolucionais para alcançar a super resolução das imagens, sendo o código treinado antes com as imagens obtidas, comparando a entrada (imagem de LR) com a saída (imagem de HR).

Tabela 2 - Exemplo de Comparação entre Imagem Original e Imagem de Saída

	Linfocito	Eosi/Basofilo	Monocito	Neutrofilo
PSNR	38.38	37.39	40.66	38.57
MSE	28.32	35.62	16.75	27.12
SSIM	0.98	0.97	0.98	0.97

Fonte: autoria própria.

Tabela 3 - Exemplo de Variância do Laplaciano das Imagens*

	Linfocito	Eosi/Basofilo	Monocito	Neutrofilo
Original	49.32	92.03	27.74	40.37
Distorcido	21.89	44.79	14.37	18.73
Previsto	23.36	50.82	15.56	19.89

* “Distorcido” refere-se a imagem de entrada e “Previsto” refere-se a imagem de saída.

Fonte: autoria própria.

O código de Super Resolução foi feito seguindo como base 3 grupos diferentes de autores. O primeiro são os autores do artigo sobre SRCNN (Dong *et al.* (2016)). O segundo é o autor do código que foi usado como referência, acessado pelo link na data 07/08/2022 (https://colab.research.google.com/github/goodboychan/chans_jupyter/blob/main/_notebooks/2020-10-13-01-Super-Resolution-CNN.ipynb#scrollTo=euWZAmczlXnm). E o autor que fez a divisão diferenciada do *dataset* acessado pelo link na data 07/08/2022 (<http://mmlab.ie.cuhk.edu.hk/projects/SRCNN.html>), que será melhor explicada nos próximos tópicos.

Assim, o código foi rodado em máquina local por necessitar de muito tempo e de uma quantidade alta de imagens, com o objetivo de fazer super resolução (melhorar a qualidade) de imagens de WBCs, sendo treinada com as mesmas. Essa SRCNN é uma rede neural de aprendizado profundo (*deep learning neural network*) que aprende *end-to-end mapping* das imagens de baixa resolução para imagens de alta resolução. A primeira atividade é preparar o *dataset* do treinamento e da validação, sendo utilizadas as imagens pós *Data Augmentation* que, a partir das imagens de HR e LR, são 17.928 de cada (que estão divididas em 3 pastas). Foram feitos 30 recortes aleatórios (baseando-se no autor do *dataset*) de tamanho 32 x 32 para as imagens de LR e de 20 x 20 para as imagens de HR (os tamanhos serão explicados posteriormente). O resultado foi um total de 321.840 imagens de treinamento em cada pasta de LR e HR (10.728 imagens na pasta de treinamento multiplicado por 30) e 110.160 imagens de validação em cada pasta de LR e HR

(3.672 imagens na pasta de validação multiplicado por 30). Desse modo, foram salvas as imagens, criando arquivos binários para serem chamados dentro do modelo (essa parte do código é rodada somente uma vez, pois são salvos os dois arquivos binários para serem sempre usados os mesmos na parte de treinamento e validação).

Na sequência, são chamadas as métricas descritas anteriormente para poder comparar quantitativamente as imagens pré e pós-SR e ajustado o fator/escala que foi usado (valor de quatro vezes). Após todos os processos anteriores, é estruturado o modelo de SRCNN, com mudanças da quantidade de camadas e também de parâmetros para descobrir a melhor estrutura de SRCNN conforme os valores das métricas. Em seguida, ocorre o carregamento do *dataset* de treinamento e de validação dentro do modelo para treinar a rede neural e, depois, o treinamento, no qual acontece o salvamento dos melhores pesos e treina com o *dataset* feito. Posteriormente a todo o processo de treinamento, são feitas as comparações, sendo divididas em comparações visuais e quantitativas. Na primeira são comparadas as imagens um ao lado da outra para observar suas diferenças e melhorias. Na segunda, são expostos os resultados alcançados das métricas escolhidas, observando-se em quais ocorrem melhores valores para obter a melhor imagem a partir de LR.

O subtópico seguinte é “Testes” que explica cada tipo de mudança feita como teste dentro do código de Super Resolução. São feitas comparações para alcançar o melhor modelo, sendo que os testes foram baseados no artigo Dong *et al.* (2016) e em alguns cursos feitos pela autora do trabalho nos quais o foco era fazer desde mudanças estruturais até em parâmetros da rede neural.

Deve-se ressaltar que a imagem de “*Input*” é diferente da imagem que somente sofre *resize up* e *down* (que seria a imagem LR), pois ocorrem alguns pré e pós-processos (comentados anteriormente) para que seja possível a entrada da imagem no modelo, para que ocorra a saída de uma imagem melhorada. Também deve ser salientado que é sempre feita a comparação entre a imagem “*Input*” (Distorcida) e a Original x a imagem “*Output*” (Prevista) e a Original.

Antes de começar a fazer os testes, foram elencados os seguintes passos:

- Rodar 3 vezes o código (e reiniciar, caso contrário, começa a treinar do momento que parou) e fazer média, para observar se mantém os resultados, caso contrário saem da curva. Há um ponto de atenção, que é rodar com o mesmo *dataset* de treino e de validação (os arquivos binários .h5) e as mesmas imagens para comparação de todas as mudanças feitas.

- Configurar o parâmetro “*Epoch*” sempre com o valor 50 (para treinar o suficiente e não demorar demais), além de somente salvar os melhores pesos e resultados do modelo quando ocorre melhoria, para não ocorrer o *OverFitting*, não sendo necessário usar o *Early Stopping*.
- Usar *dataset* do “*Test*” para recriar as imagens em SR, sendo que não foram usadas antes para treinar no modelo. Assim, usar as pastas divididas (treinamento com 60%, validação com 20% e teste com 20%).
- Fazer testes de cada tipo de imagem (cada tipo de célula (4), escolher cada uma aleatória dentro da pasta e se manter com ela em todos os testes), pois como existem menos amostras de certas classes, pode ocorrer que a imagem não melhore tão bem quanto as outras que possuem mais imagens.
- Fazer “média” entre as imagens, pois talvez cada tipo de imagem diferente precise de CNNs diferentes, mas seria muito trabalhoso. Em seguida, verificar quais possuem melhores resultados entre todas as imagens testadas.
- Salvar em todos os testes os *prints* dos modelos, dos resultados finais de cada célula (escolher uma célula e refazer o teste 3 vezes e fazer a média, para verificar se mantém os mesmos resultados), dos sumários dos modelos, e também a própria pasta com o *input* e *output* e “SRCNN” (contendo *checkpoints*, variáveis, pesos, entre outros).

Com tudo configurado, a primeira mudança foi a quantidade de camadas convolucionais. Porém, o código não tem a “estrutura” normal de CNN, pois não há necessidade de aplicar *Pooling* e *Fully Connected Layers (Flatten + Dense)*. Como o resultado se transforma em uma nova imagem, não é preciso existirem certas camadas (como *pooling*, *dense* e *flatten*), porque são para diminuir o tamanho da imagem, transformar em classes ou pertencer a um grupo. Assim, o código de Super-Resolução não resulta em um número nem em binário, o resultado transforma-se em uma nova imagem, como observado no artigo em que foi fundamentado (Dong *et al.*, 2016).

Ainda, conforme o *dataset* foi preparado de forma a ter as imagens de LR com 32 x 32 e o HR tem 20 x 20, as operações dentro das camadas convolucionais precisam ajustar esse formato para que fiquem, no final, 20 x 20 ambas e possa ocorrer o melhoramento da imagem (no caso a imagem LR sai do tamanho de 32 x 32 e alcança o tamanho 20 x 20). Esse formato é alcançado pelo “*kernel size*” quando

usado juntamente com o parâmetro ajustado no “Valid”, que faz o *output* não ter o mesmo tamanho do *input*. O *kernel size* possui a equação descrita na Equação 3.

Equação 3 – kernel size ajustado em Valid

$$1 + \frac{(Height_{input} + 2 * pad - kernelSize)}{stride} \quad (4)$$

Fonte: autoria própria.

Assim, por exemplo, o valor 13 foi usado na primeira camada como *kernel size* juntamente com *padding* “Valid” para que 32 x 32 alcance 20 x 20. Substituindo os valores, a “Height Input” (altura da imagem de entrada) é 32 com valor do “pad” igual a 0, pois ele não mantém o mesmo tamanho do *input*, “kernel size” com 13 e o “stride” igual a 1, atingimos o tamanho de 20 x 20 da imagem de saída, como mostrado na Equação 4.

Equação 4 - Exemplo de *kernel size* ajustado em Valid com valores

$$1 + \frac{32 + 2 * 0 - 13}{1} = 20 \quad (5)$$

Fonte: autoria própria.

Nas próximas camadas, não é obrigatório usar o valor “13” nem o parâmetro configurado em “valid”, pois existem mais camadas e o processo pode ser “distribuído”, porque a fórmula que está na Equação 4 é para cada camada dentro da CNN. Porém, é preciso que, com os ajustes entre as camadas, os valores finais saiam de 32 para 20.

Assim, com os testes feitos com o objetivo de mudar a quantidade de camadas convolucionais (vide Figura 26), foi possível observar que houve um aumento de uma camada, iniciando em uma camada até alcançar cinco camadas, sendo que a cada camada adicionada, dobrou-se a quantidade de filtros/features maps. No primeiro modelo, tem-se uma camada com 8 features maps com kernel size de 13 e o parâmetro ajustado em “valid”, mostrando também o tempo estimado por “epoch” e o tempo total contabilizando 50 epochs. Já no segundo modelo, existem duas camadas, sendo a primeira igual ao modelo anterior e a segunda com o valor de features maps dobrado e com kernel size 5, para manter o padrão e somente ter a mudança na quantidade de camadas, preservando todos os outros parâmetros iguais, somente

analisando essa mudança específica. Desse modo, mantendo as proporções e os parâmetros até atingir cinco camadas.

Figura 26 - Super Resolução - Mudança - Mudanças das Camadas

```
Testes:
{8C13(valid)} (tempo = 45 s/epoch => 37,5 min)
{8C13(valid)} --> {16C5(same)} (tempo = 1 min e 30 s/epoch => 75 min)
{8C13(valid)} --> {16C5(same)} --> {32C5(same)} (tempo = 2 min e 50 s/epoch => 2 horas e 21 min)
{8C13(valid)} --> {16C5(same)} --> {32C5(same)} --> {64C5(same)} (tempo = 5 min e 40 s/epoch => 4 horas e 43 min)
{8C13(valid)} --> {16C5(same)} --> {32C5(same)} --> {64C5(same)} --> {128C5(same)} (tempo = 16 min e 15 s/epoch => 13 horas e 30 min)
```

Fonte: autoria própria.

A segunda mudança envolveu os filtros ou *features maps* que, a partir dos testes feitos anteriormente, empregaram a estrutura de camadas do Modelo 5, por ter apresentado os melhores resultados. Os testes começaram com a mesma estrutura que foi “vencedora” da categoria passada. Todos os 5 modelos mantêm as cinco camadas, porém a camada inicial começa com o dobro de *features maps* que a anterior tinha, e foi aumentando com o passar das camadas, como é possível observar na Figura 27.

A terceira mudança foi adicionar a estrutura “*Dropout*” no modelo, o qual é uma técnica que elimina aleatoriamente (e temporariamente) alguns dos neurônios ocultos na rede. É muito utilizado para ajudar no problema do “*Overfitting*” e generalizar mais a rede, como mostra a Figura 28, a qual foi adicionada a partir do segundo modelo cinco camadas de *Dropout* de 10% entre as camadas convolucionais, e que aumentou em 10% a cada modelo até alcançar o último.

Figura 27 - Super Resolução - Mudança - Filtros

```
Testes:
{8C13(valid)} --> {16C5(same)} --> {32C5(same)} --> {64C5(same)} --> {128C5(same)} (mesmo que o anterior, copiar a pasta) (+- 13 horas)
{16C13(valid)} --> {32C5(same)} --> {64C5(same)} --> {128C5(same)} --> {256C5(same)} (+- 42 horas)
{32C13(valid)} --> {64C5(same)} --> {128C5(same)} --> {256C5(same)} --> {512C5(same)} (+- 150 horas) ou 6 dias e 6 horas
{64C13(valid)} --> {128C5(same)} --> {256C5(same)} --> {512C5(same)} --> {1024C5(same)} (cancelado pelo tempo)
{128C13(valid)} --> {256C5(same)} --> {512C5(same)} --> {1024C5(same)} --> {2048C5(same)} (cancelado pelo tempo)
```

Fonte: autoria própria.

Figura 28 - Super Resolução - Mudança - Dropout

```

Testes: (7 dias)
{8C13(valid)} --> {16C5(same)} --> {32C5(same)} --> {64C5(same)} --> {128C5(same)}
{8C13(valid)-D10} --> {16C5(same)-D10} --> {32C5(same)-D10} --> {64C5(same)-D10} --> {128C5(same)-D10}
{8C13(valid)-D20} --> {16C5(same)-D20} --> {32C5(same)-D20} --> {64C5(same)-D20} --> {128C5(same)-D20}
{8C13(valid)-D30} --> {16C5(same)-D30} --> {32C5(same)-D30} --> {64C5(same)-D30} --> {128C5(same)-D30}
{8C13(valid)-D40} --> {16C5(same)-D40} --> {32C5(same)-D40} --> {64C5(same)-D40} --> {128C5(same)-D40}

```

Fonte: autoria própria.

A quarta mudança é trocar certos parâmetros, podendo ser diferentes tipos como, começando com a Função de ativação ou *Activation Function*, sendo escolhido cinco funções diferentes, para observar o efeito de cada uma, sendo elas: Relu, Elu, Selu, Sigmoid e TanH.

A mudança seguinte foi trocar os valores do *kernel size* e do *Padding*, pois é possível observar que a primeira camada está com 13 x 13 ajustado em “Valid”, como explicado anteriormente. Seria interessante observar as diferenças nos resultados retirando os valores 13 x 13 da primeira camada para a segunda (como mostra a Figura 29), terceira, quarta e quinta camadas e, após, também fazer a “distribuição” do 13 x 13, seguindo o cálculo já demonstrado na Equação 3. Assim, colocando em duas, três, quatro e cinco camadas seguidas, como mostra a Figura 30.

A próxima mudança foi trocar o valor booleano do parâmetro “Bias” entre verdadeiro (*True*) e falso (*False*) como mostra a Figura 31.

Figura 29 - Exemplo de Recolocação do 13 x 13 nas Camadas 1 e 2

```

def model():
    SRCNN = tf.keras.Sequential(name='SRCNN')
    SRCNN.add(tf.keras.layers.Conv2D(filters=8, kernel_size=(13, 13),
                                     padding='VALID',
                                     use_bias=True,
                                     input_shape=(None, None, 1),
                                     kernel_initializer='glorot_uniform',
                                     activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=16, kernel_size=(5, 5),
                                     padding='SAME',
                                     use_bias=True,
                                     kernel_initializer='glorot_uniform',
                                     activation='relu'))

def model():
    SRCNN = tf.keras.Sequential(name='SRCNN')
    SRCNN.add(tf.keras.layers.Conv2D(filters=8, kernel_size=(5, 5),
                                     padding='SAME',
                                     use_bias=True,
                                     input_shape=(None, None, 1),
                                     kernel_initializer='glorot_uniform',
                                     activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=16, kernel_size=(13, 13),
                                     padding='VALID',
                                     use_bias=True,
                                     kernel_initializer='glorot_uniform',
                                     activation='relu'))

```

Fonte: autoria própria.

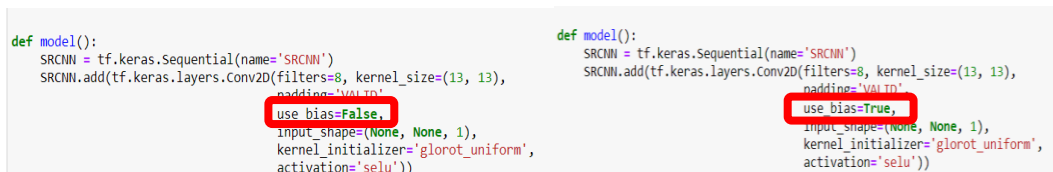
Figura 30 - Exemplo de Distribuição do 13 x 13 nas Camadas 2 e 3



The figure displays two code snippets for a Keras model named 'SRCNN'. In the left snippet, a red box highlights the configuration of the first Conv2D layer: `filters=8, kernel_size=(7, 7), padding='VALID', use_bias=True, input_shape=(None, None, 1), kernel_initializer='glorot_uniform', activation='relu'`. In the right snippet, a red box highlights the configuration of the second and third Conv2D layers: `filters=16, kernel_size=(5, 5), padding='VALID', use_bias=True, kernel_initializer='glorot_uniform', activation='relu'` and `filters=32, kernel_size=(5, 5), padding='VALID', use_bias=True, kernel_initializer='glorot_uniform', activation='relu'`.

Fonte: autoria própria.

Figura 31 - Modelos com os valores de Bias com True e False



The figure displays two code snippets for a Keras model named 'SRCNN'. In the left snippet, a red box highlights `use_bias=False` in the Conv2D layer configuration. In the right snippet, a red box highlights `use_bias=True` in the Conv2D layer configuration. Both snippets show the same layer configuration: `filters=8, kernel_size=(13, 13), padding='VALID', use_bias=True, input_shape=(None, None, 1), kernel_initializer='glorot_uniform', activation='selu'`.

Fonte: autoria própria.

A próxima mudança foi substituir o valor do *Kernel Initializer* por três diferentes: *Glorot Uniform*, *Random Uniform* e *He Uniform*.

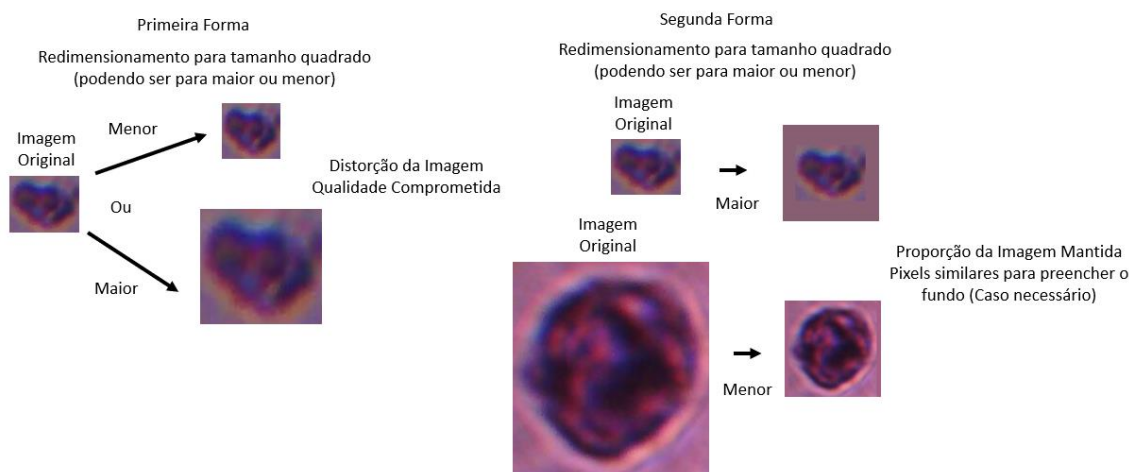
A outra troca foi alterar o parâmetro do Otimizador entre: Adam, SGD e Ftrl. Entretanto, ocorreu o mesmo problema da criação de um quadro em volta da imagem com os otimizadores SGD e Ftrl.

A última mudança foi alterar o parâmetro da Taxa de Aprendizado (*Learning Rate* - LR) entre valores que possuem diferentes casas decimais, como: 0,00005; 0,0003 e 0,005. A alteração foi para observar a diferença que a alteração nas casas decimais podem causar nos resultados.

Após a criação das imagens, sua validação e escolhendo o melhor modelo para gerar as melhores imagens de SR, foi feito código de classificação das imagens. Entretanto, como o código de classificação também é uma CNN, é necessário que as imagens sejam quadradas (mesma largura e altura). Assim, foram articuladas algumas formas para resolver esse problema. A primeira forma é fazer o "Resized" da imagem (podendo ser *up* ou *down*, não possuindo um valor específico, sendo que o ideal seria observar qual padrão se repete mais para depois aplicar nas outras imagens, adaptando o restante). Porém, ocorre a distorção da imagem, perdendo também a qualidade. O segundo modo é criar uma solução em que é possível transformar a imagem em quadrada sem distorcer as proporções da imagem original,

preenchendo o fundo com *pixels* similares. Assim, preserva as dimensões do tamanho original ou as proporções da imagem dentro da imagem quadrada recriada, como mostra a Figura 32.

Figura 32 - Exemplos de modelos para gerar imagens com Super Resolução



Fonte: autoria própria.

Portanto, utilizou-se a segunda opção, sendo criadas duas funções. A primeira é responsável pelas imagens menores do que o tamanho pedido, centralizando a imagem do meio e preenchendo o resto do campo com *pixels* de cores similares às das bordas. Mantém-se, desse modo, exatamente o mesmo tamanho que estava originalmente, somente sendo adicionado os *pixels* de fundo para que não fique o fundo com a cor preta. Já a segunda função é responsável pelas imagens maiores do que o tamanho pedido, necessitando fazer o redimensionamento mantendo as proporções para o tamanho menor (máximo que possível) e o fundo preenchido com *pixels* de cores similares às das bordas. Desse modo, são criadas imagens todas quadradas e sem distorção, sendo de extrema importância para o treinamento das Redes Neurais.

Após criar as funções de redimensionamento, foi necessário criar um *dataset* específico para o treinamento de classificação de cada tipo de célula, pois como o *dataset* foi feito para treinar o código de Super-Resolução, são trinta cortes aleatórios feitos dentro de cada imagem. Assim, não poderia ser utilizado por ser necessário ter a célula por completo em cada imagem para que sejam extraídas suas características e, em seguida, aprendidas pela Rede Neural. E, por isso, fez-se necessária a criação das duas funções de redimensionamento para configurar o *dataset*.

Foram utilizadas as 249 imagens originais das células, sendo 195 neutrófilos, 17 monócitos, 33 linfócitos e 4 eosinófilos/basófilos e cada tipo de célula com quantidades diferentes. Isso representa uma quantidade ínfima para ocorrer o treinamento de uma Rede Neural. Consequentemente, houve também a necessidade de equalizar a quantidade de imagens por tipo de célula. Desse modo, o *Data Augmentation* de cada tipo de classe de células sofreu mudanças diferentes e em quantidades diferentes.

Para os eosinófilos/basófilos que possuem uma quantidade menor, foram feitas duas mudanças: de rotação de um grau em um grau (360 mudanças) e as mesmas rotações com *flipping* (mais 360 mudanças). Portanto, 4 vezes 720 dando um total de 2.880 imagens feitas, continuando uma quantidade baixa de imagens. Em posse dessas 2.880 imagens, foram feitas mais mudanças de contraste (pouco, médio e alto) e de brilho (pouco, médio e alto), no total de 6 transformações das 2.880 imagens, resultando em 17.280 imagens.

Já nos monócitos, foram feitas também mudanças de rotação e de rotação e *flipping*, somente a diferença sendo de 4 em 4 graus com 90 giros, sendo 180 giros para cada uma das 17 imagens, criando 3.060 imagens. Desse modo, com essas 3.060 imagens foram feitas novamente mudanças de contraste (pouco, médio e alto) e de brilho (pouco, médio e alto), no total de 6 transformações das 3.060 imagens, resultando em 18.360 imagens.

Para os neutrófilos, que possuem 195 imagens, foram necessários menos tipos de transformações. Logo, foram feitas rotações e rotações com *flipping* de 8 em 8 graus com 45 giros, totalizando 90 giros para cada uma das 195 imagens. Completou-se um total de 17.550 imagens, não precisando de transformações que mudassem o contraste ou o brilho das imagens.

Por último, para os linfócitos, foi feita a mesma mudança da classe dos neutrófilos, com um adicional de funções de contraste (pouco, médio e alto) e de brilho (pouco, médio e alto). Nas 33 imagens originais, foram feitos 90 giros, resultando em 2970 imagens em que cada uma delas recebeu seis funções diferentes de contraste e de brilho, apresentados anteriormente, completando um total de 17.820 imagens.

Em seguida, foi feita a separação das imagens para as pastas de treinamento e de teste, sendo 70% e 30%, respectivamente. Desse modo, realizaram-se as mudanças necessárias, como normalização, transformação em vetores, embaralhamento das variáveis para que fosse possível rodar tanto na CNN de

classificação, como também na SVM fornecida pela Hilab, para que fosse possível uma comparação entre os desempenhos dos classificadores.

Na parte dos testes de ambos os códigos, foram extraídos os valores das métricas já propostas na seção Hipótese, que são acurácia, sensibilidade, especificidade e precisão, sendo também adicionada a F1-score para mais credibilidade (todas atingindo um mínimo de 85%). Cada métrica representa um valor de parâmetro do código, sendo definidas por Claro *et al.* (2020); Quan, Wang e Liu (2020) por equações definidas nas Equações 5, 6, 7, 8 e 9.

Equação 5 – Acurácia

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

Fonte: autoria própria.

Equação 6 – Sensibilidade

$$Sensitivity = \frac{TP}{TP + FN} \quad (7)$$

Fonte: autoria própria.

Equação 7 – Especificidade

$$Specificity = \frac{TN}{TN + FP} \quad (8)$$

Fonte: autoria própria.

Equação 8 – F1_score

$$F1_score = \frac{2TP}{2TP + FP + FN} \quad (9)$$

Fonte: autoria própria.

Equação 9 – Precisão

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

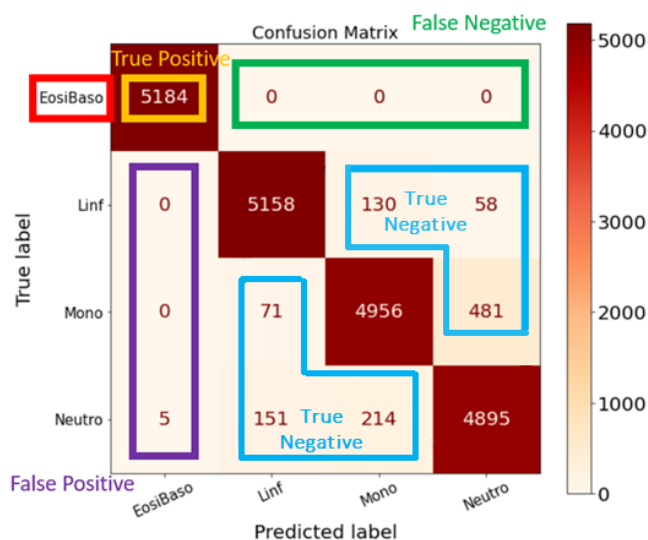
Fonte: autoria própria.

Assim, a acurácia (*accuracy* ou ACC) que avalia o percentual de acertos, foi obtida pela razão entre a quantidade de acertos e o total de entradas. Já a sensibilidade (*sensitivity*), conhecida também como *recall* ou revocação, avalia a

competência de detectar os resultados classificados como positivos. Por outro lado, a métrica de especificidade (*specificity*) avalia a capacidade de detectar os resultados negativos. A precisão (*precision*) avalia a quantidade de verdadeiros positivos sobre a soma de todos os valores positivos. E *F1_score*, é uma média harmônica calculada com base na precisão e na sensibilidade.

De acordo com as referências anteriores, TP (*True Positive*) ou Verdadeiro Positivo, que, no caso, significa que foi prevista a imagem para determinada classe e realmente são da classe, por exemplo, a imagem foi prevista ser um linfócito e de fato, é um linfócito. Já TN (*True Negative*) ou Verdadeiro Negativo, a imagem não é de determinado grupo e realmente não é do grupo. O FP (*False Positive*) ou Falso Positivo, é quando a imagem prevista é de um tipo de classe, mas não é daquela classe. E, por fim, os FN (*False Negative*) ou Falso Negativo, ocorrem quando a imagem não é de determinado grupo, entretanto, de fato, é daquela classe. Além de também ser demonstrada a matriz de confusão (no caso do classificador da CNN), que mostra quais foram os acertos e os erros de cada classe feito pelo classificador, sendo possível observar TP, TN, FP e FN na própria matriz, mostrada na Figura 33.

Figura 33 - Exemplo de Matriz de Confusão



* *True Negative* – *Falso Negativo*/ *True Positive* – Verdadeiro Positivo / *False Negative* – Falso Negativo/ *False Positive* – Falso Positivo

Fonte: autoria própria.

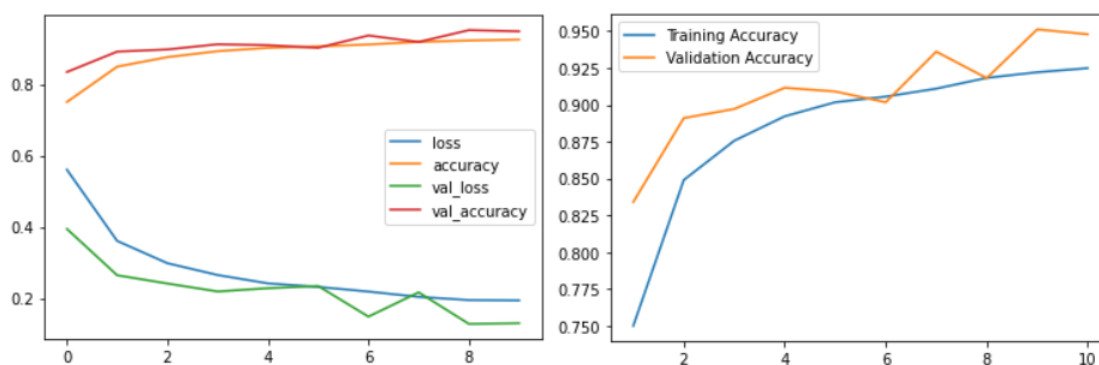
Por exemplo, analisando a classe de Eosinófilos/Basófilos (sinalizada com um retângulo vermelho), o primeiro valor que aparece é o Verdadeiro Positivo (retângulo

laranja), já os outros valores da mesma linha, são os Falsos Negativos (retângulo verde), os valores da mesma coluna são os Falsos Positivos (retângulo roxo) e o resto da tabela (retângulo azul) são os Verdadeiros Negativos. É possível observar que a maioria das classes obteve um excelente desempenho, somente as classes dos monócitos e neutrófilos que tiveram um resultado menor comparado com as outras, porém, mesmo assim, o resultado de classificação foi ótimo.

Assim, como também foi feita a classificação de uma imagem aleatória para observar os valores de probabilidade de cada classe, como também a soma de todas, com o *index* e a etiqueta da classe e também plotando a porcentagem em colunas.

Também foram plotados gráficos que demonstram a aprendizagem do algoritmo classificador, podendo serem observadas as curvas da acurácia e de perda, tanto de treinamento quanto de validação, demonstrado na Figura 34.

Figura 34 - Exemplos de Gráficos de Aprendizagem de Acurácia e de Perda



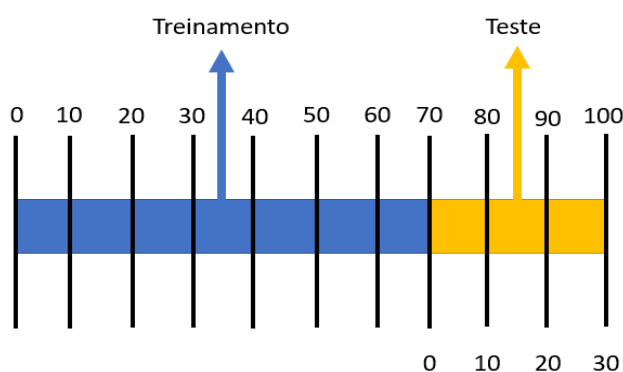
Fonte: autoria própria.

Após a configuração do código, foi testado qual seria o melhor tamanho (quadrado) para entrar nos classificadores, para obter melhor desempenho de ambos, sem ocorrer o *timeout* ou *overload* do Google Colab, sendo feitos testes em 3 tipos de formatos:

- Imagens de 70 x 70 *pixels*, sendo menor que a média padrão de todas as imagens dentro do *dataset*;
- Imagens de 85 x 85 *pixels*, sendo aproximadamente a média do tamanho das imagens que existem dentro do *dataset*;
- Imagens de 100 x 100 *pixels*, sendo maior que a média das imagens dentro do *dataset*.

Após o teste de imagem, foram feitos testes alterando as imagens do *dataset* de treinamento e de validação (por imagens de SR) para observar se ocorre alteração dos valores das métricas. O primeiro, em ir “substituindo” o *dataset* normal por imagens de SR (aumentando em porcentagem). O segundo teste, em vez de substituir, é ir adicionando ao *dataset* normal as imagens de SR, também em porcentagem. Primeiramente, o *dataset* foi dividido de uma forma fácil de entender e utilizar, como 70% do *dataset* total foi transferido para o treinamento e 30% para o teste. A divisão está mostrada na Figura 35, na qual observa-se a separação do *dataset* de treinamento em 7 partes iguais e o *dataset* de teste em 3 partes iguais, para ocorrer a substituição ou adição com facilidade e praticidade. Assim, em cada divisão ficaram separados 1.728 eosinófilos, 1.782 linfócitos, 1.836 monócitos e 1.755 neutrófilos.

Figura 35 - Divisão do *dataset*

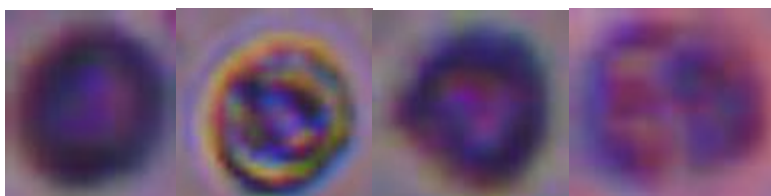


Fonte: autoria própria.

Entretanto, antes de fazer essas duas alternativas, foi feito mais um teste com a CNN (uma vez que já foi validada, pois foi comparado o desempenho com classificador simples já utilizado pela empresa Hilab e, também, focando somente no trabalho feito). Com o objetivo de observar como o classificador, com *dataset* normal e de SR, desempenha em classificar quatro tipos de imagens de célula de cada classe (as imagens utilizadas são da pasta “Output” da pasta “Melhor Resultado”) com LR, e depois testar essas mesmas imagens, porém com SR (no total 8 imagens, 4 normais e 4 com SR). Observando qual classificador (treinado com ou sem SR) apresenta o melhor resultado, consegue-se melhor classificar imagens que são de LR ou de SR.

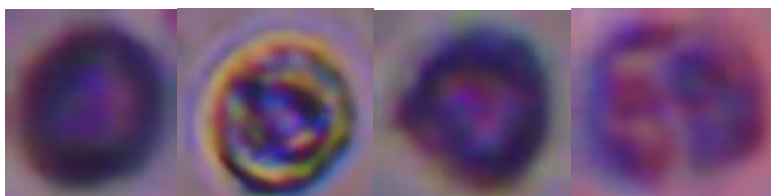
Após observar os valores das métricas, foram feitos os testes das imagens que são de LR e de SR com um exemplo de cada classe. A Figura 36 mostra as imagens de LR e a Figura 37 mostra as imagens de SR (sendo sempre essas mesmas imagens usadas em todos os testes feitos). Nas figuras, da esquerda para direita: linfócitos, eosinófilos, monócitos e neutrófilos.

Figura 36 - Imagens de LR de diferentes tipos de células



Fonte: autoria própria.

Figura 37 - Imagens de SR (4x) de diferentes tipos de células



Fonte: autoria própria.

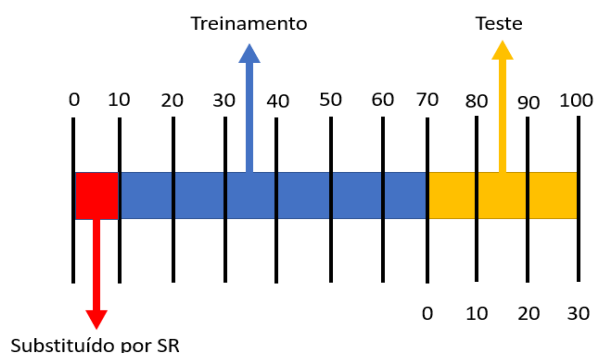
Esse tipo de teste tem como objetivo analisar se o algoritmo consegue classificar melhor uma imagem de LR ou de SR, com as mudanças do *dataset* de treinamento e de validação. Como existem laboratórios com microscópios com lentes de menor valor, que se tivessem um algoritmo de SR poderiam aumentar as chances de acertar certos tipos de células, não seria preciso investir em um equipamento mais caro e com lentes custosas.

A primeira tentativa foi no *dataset* 100% original, sem nenhuma imagem com SR. Já a segunda tentativa foi no *dataset* 100% com imagens de SR.

Assim, após a comparação com *dataset* 100% de imagens originais e 100% com imagens de SR, foi feito o teste das duas alternativas mencionadas anteriormente. Dessa forma, foram separados 5 testes para cada conjunto de imagens para testar a "substituição" e "adição". O primeiro teste (para a substituição) foi trocar o primeiro "bloco" pelas imagens que foram feitas através da SR (são as mesmas

imagens que estavam nesse primeiro bloco, porém em vez de serem as originais, são as de SR), como mostra a Figura 38.

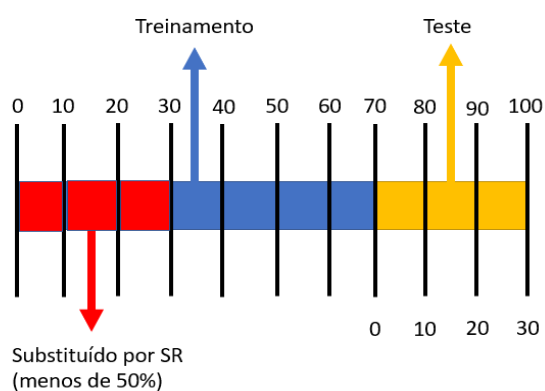
Figura 38 - Primeira mudança de substituição do *dataset*



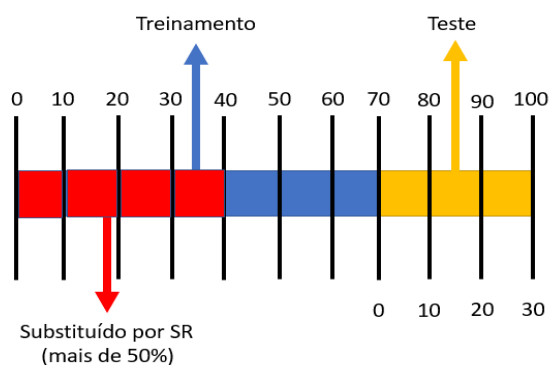
Fonte: autoria própria.

O segundo e terceiro testes focaram em testar com menos ou mais de 50% de troca do *dataset* de treinamento original com as imagens de SR. A Figura 39 mostra com menos de 50% e na Figura 40 demonstra com mais de 50% do *dataset* de treinamento.

Figura 39 - Segunda mudança de substituição do *dataset*

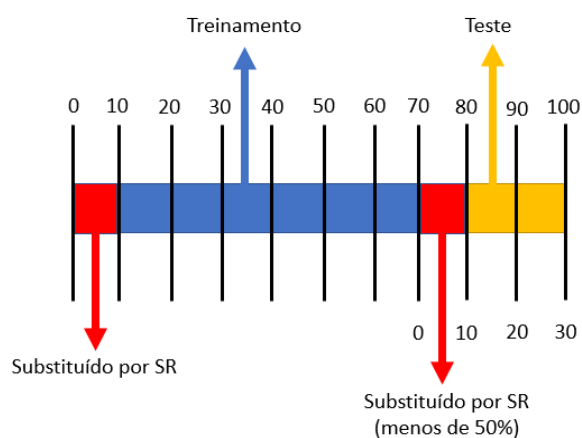


Fonte: autoria própria.

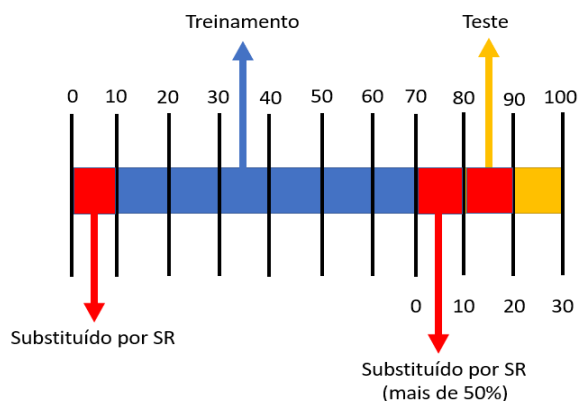
Figura 40 - Terceira mudança de substituição do *dataset*

Fonte: autoria própria.

Dessa maneira, após analisar os três testes, foram feitos mais dois testes completando os cinco testes vistos anteriormente. Só que em vez de mudar somente o *dataset* de treinamento, foi mudado também o dataset de validação, sendo os dois últimos testes. O primeiro teste muda menos de 50% do *dataset* de validação, aproximadamente 33% e o segundo muda mais de 50% do *dataset* de validação, aproximadamente 66%. Como o primeiro teste (do primeiro algoritmo, que mudou somente um “bloco” do *dataset* de treinamento) foi que apresentou os melhores resultados (como será mostrado na próxima seção), foi utilizado o mesmo com duas mudanças no *dataset* de validação, sendo o quarto teste apresentado na Figura 41 e o quinto teste apresentado na Figura 42.

Figura 41 - Quarta mudança de substituição do *dataset*

Fonte: autoria própria.

Figura 42 - Quinta mudança de substituição do *dataset*

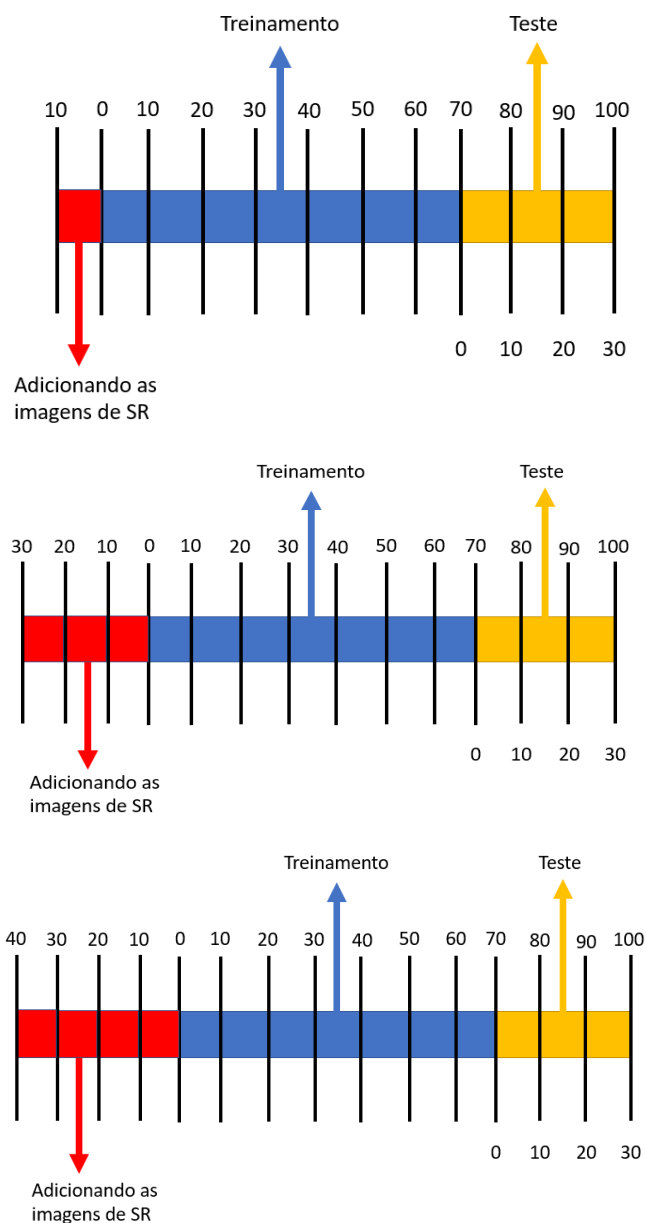
Fonte: autoria própria.

Com o quarto teste está a primeira mudança mais a mudança do primeiro “bloco” do *dataset* de validação, para testar se com a modificação do *dataset* de validação, ocorre melhora das métricas de classificação.

Já no quinto teste, além de manter a modificação da primeira mudança e também do primeiro “bloco” do *dataset* de validação, aconteceu a alteração do segundo “bloco”. Isso foi devido ao quarto teste estar com uma mudança de 33% e o quinto teste com 66% para observar qual mudança (se com menos ou mais de 50%) seria mais efetiva.

Após os testes com substituição, foram feitas as mudanças de “adição” que, em vez de substituir como foi realizado anteriormente, adicionaram as imagens de SR dentro do *dataset* de treinamento (Figura 43). As mudanças de adição foram feitas no mesmo formato das mudanças de substituição (primeira mudança acrescentou um “bloco”, a segunda modificação aumentou três “blocos”, com finalidade de ser menos de 50% de mudança, a terceira alteração incluiu quatro “blocos”, para ser mais de 50% de modificação).

Figura 43 - Primeira, segunda e terceira mudanças de adição do *dataset*

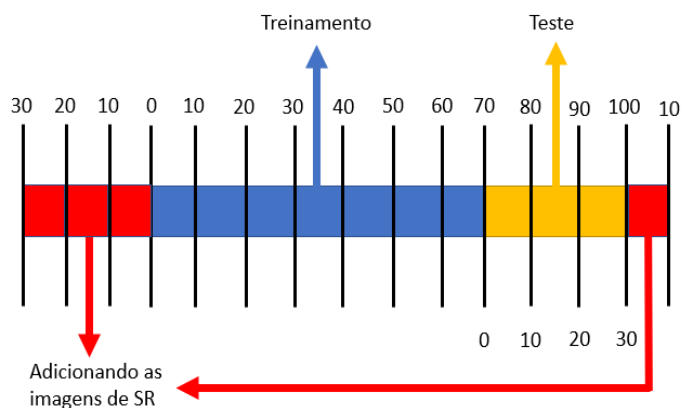


Fonte: autoria própria.

Assim, depois dos três testes, como nos testes de substituição, foram realizados mais dois testes. O quarto (menos de 50% de adição, no caso 33%) e quinto (mais de 50% de adição, no caso 66%) nos mesmos moldes, que em vez de substituir, adiciona “blocos” de imagens no *dataset* de validação, completando os cinco testes explicados previamente, com a quarta mudança ilustrada Figura 44 e quinta apresentada na Figura 45.

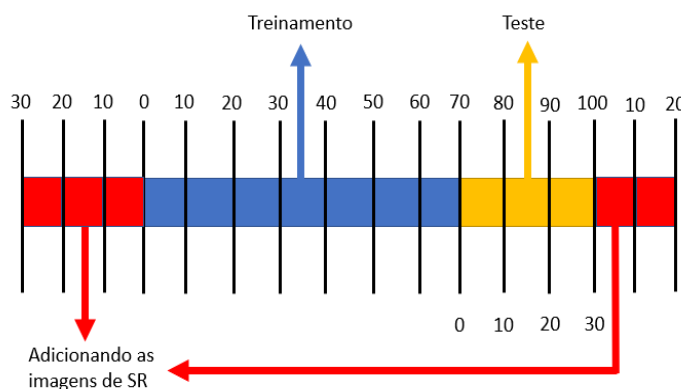
Dado que a segunda mudança obteve os melhores resultados comparada com a primeira e a terceira mudanças, foi usado o mesmo para fazer o quarto e o quinto testes.

Figura 44 - Quarta mudança de adição do *dataset*



Fonte: autoria própria.

Figura 45 - Quinta mudança de adição do *dataset*



Fonte: autoria própria.

A quarta mudança adicionou um “bloco” no *dataset* de validação. Já a quinta mudança acrescentou dois “blocos” no *dataset* de validação, ambos juntamente com a segunda mudança, com objetivo de observar qual modificação (se com menos ou mais de 50% do *dataset* de validação) é mais efetiva na melhora das métricas de classificação.

Assim, finalizando toda a descrição da metodologia feita para os testes, a próxima seção apresenta os resultados encontrados.

4 RESULTADOS

4.1 RESULTADOS ALCANÇADOS

De todas as mudanças das estruturas com camadas, registrando os valores em uma tabela Excel mostrado na Tabela 4, a que obteve melhor resultado foi registrado na Tabela 05 é o modelo com cinco camadas (de acordo com as médias feitas, conseguiu melhor colocação em todas as métricas). Com os resultados em toda essa seção, compre-se o objetivo específico 4.

Tabela 4 - Resultados da Mudança das Camadas

Resultado 01												
Linfócito			Eosi/Basófilo			Monócito			Neutrófito			
PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM	
Original x Distorcido	37,01	38,85	0,97	34,49	69,37	0,95	39,32	22,83	0,97	37,39	35,61	0,96
Original x Previsto	38,38	28,32	0,98	37,39	35,62	0,97	40,66	16,75	0,98	38,57	27,12	0,97
Resultado 02												
Linfócito			Eosi/Basófilo			Monócito			Neutrófito			
PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM	
Original x Distorcido	37,01	38,85	0,97	34,49	69,37	0,95	39,32	22,83	0,97	37,39	35,61	0,96
Original x Previsto	38,30	28,85	0,98	37,26	36,67	0,97	40,53	17,27	0,98	38,36	28,45	0,97
Resultado 03												
Linfócito			Eosi/Basófilo			Monócito			Neutrófito			
PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM	
Original x Distorcido	37,01	38,85	0,97	34,49	69,37	0,95	39,32	22,83	0,97	37,39	35,61	0,96
Original x Previsto	38,28	28,99	0,98	37,22	36,98	0,97	40,48	17,45	0,98	38,34	28,60	0,97
Média												
Linfócito			Eosi/Basófilo			Monócito			Neutrófito			
PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM	
Média_Original x Distorcido	37,01	38,85	0,97	34,49	69,37	0,95	39,32	22,83	0,97	37,39	35,61	0,96
Média_Original x Previsto	38,32	28,72	0,98	37,29	36,42	0,97	40,56	17,16	0,98	38,42	28,06	0,97
Média todos - PSNR	38,65											
Média todos - MSE	27,59											
Média todos - SSIM	0,98											
1 Camada												
Linfócito			Eosi/Basófilo			Monócito			Neutrófito			
R_01	R_02	R_03	R_01	R_02	R_03	R_01	R_02	R_03	R_01	R_02	R_03	
Original	49,32	49,32	49,32	92,03	92,03	92,03	27,74	27,74	27,74	40,37	40,37	40,37
Distorcido	21,89	21,89	21,89	44,79	44,79	44,79	14,37	14,37	14,37	18,73	18,73	18,73
Previsto	23,36	23,47	23,65	50,82	50,79	50,67	15,56	15,51	15,55	19,89	19,79	19,82
D	P		D	P		D	P		D	P		
Média todos - Var Lao	21.89	23.49		44.79	50.76		14.37	15.54		18.73	19.84	

Fonte: autoria própria.

Na Tabela 5, estão os resultados das médias dos valores obtidos em cada tipo de mudança de camadas e por apresentar melhores resultados, a com cinco camadas foi a escolhida.

Tabela 5 - Resultado Final da Mudança de Camadas

	Quantidade de Camada/s				
	1	2	3	4	5
PSNR	38,648	38,665	38,771	38,826	39,032
MSE (menor valor - melhor)	27,588	27,394	26,717	26,333	25,026
SSIM	0,975	0,976	0,9765	0,9771	0,9774
Var_Lap - L	23,494	22,926	23,674	23,530	23,709
Var_Lap - E/B	50,758	50,260	52,219	52,723	58,403
Var_Lap - M	15,540	15,467	15,794	15,855	16,020
Var_Lap - N	19,836	19,881	19,805	20,048	20,621

Fonte: autoria própria.

A Figura 46 mostra o modelo que demonstrou o melhor resultado.

Figura 46 - Melhor Modelo com Mudanças de Camadas

```
def model():
    SRCNN = tf.keras.Sequential(name='SRCNN')
    SRCNN.add(tf.keras.layers.Conv2D(filters=8, kernel_size=(13, 13),
                                      padding='VALID',
                                      use_bias=True,
                                      input_shape=(None, None, 1),
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=16, kernel_size=(5, 5),
                                      padding='SAME',
                                      use_bias=True,
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=32, kernel_size=(5, 5),
                                      padding='SAME',
                                      use_bias=True,
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=64, kernel_size=(5, 5),
                                      padding='SAME',
                                      use_bias=True,
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=128, kernel_size=(5, 5),
                                      padding='SAME',
                                      use_bias=True,
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
```

Fonte: autoria própria.

Com esse resultado, é possível observar que aumentando a quantidade de camadas, ou a complexidade computacional, ocorreu uma melhora significativa nos valores das métricas, ajudando o modelo a trazer melhores resultados.

De todas as mudanças das estruturas com *features maps*, registrando os valores em uma tabela Excel, a que obteve melhor resultado entre as 3 primeiras foi (de acordo com as medidas retiradas, conseguiu melhor colocação na maioria das

métricas), registrado na Tabela 06 é o modelo que inicia com 8 *features maps* até alcançar 128 *features maps* na última camada.

Na Tabela 06, estão os resultados das médias dos valores obtidos em cada tipo de mudança de *features maps* e por apresentar melhores resultados, a que começa com 8 *features maps* foi a escolhida.

Tabela 6 - Resultado Final da Mudança dos *Features Maps*

	Quantidade de Features maps/Filtros				
	8	16	32	64	128
PSNR	39,032	38,958	38,687	0,000	0,000
MSE (menor valor - melhor)	25,026	25,598	27,336	0,000	0,000
SSIM	0,977	0,977	0,977	0,000	0,000
Var_Lap - L	23,709	23,913	23,702	0,000	0,000
Var_Lap - E/B	58,403	58,619	54,534	0,000	0,000
Var_Lap - M	16,020	15,954	15,909	0,000	0,000
Var_Lap - N	20,621	20,615	20,635	0,000	0,000

Fonte: autoria própria.

A Figura 47 mostra o modelo que demonstrou o melhor resultado.

Figura 47 - Melhor Modelo com Mudanças dos *Features Maps*

```
def model():
    SRCNN = tf.keras.Sequential(name='SRCNN')
    SRCNN.add(tf.keras.layers.Conv2D(filters=8, kernel_size=(13, 13),
                                      padding='VALID',
                                      use_bias=True,
                                      input_shape=(None, None, 1),
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=16, kernel_size=(5, 5),
                                      padding='SAME',
                                      use_bias=True,
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=32, kernel_size=(5, 5),
                                      padding='SAME',
                                      use_bias=True,
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=64, kernel_size=(5, 5),
                                      padding='SAME',
                                      use_bias=True,
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=128, kernel_size=(5, 5),
                                      padding='SAME',
                                      use_bias=True,
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
```

Fonte: autoria própria.

Com esse resultado é possível observar, que mesmo elevando a complexidade e o custo computacional, não houve maior melhora, mantendo o mesmo modelo anterior com os mesmos parâmetros.

De todas as mudanças das estruturas com *Dropout*, a que obteve melhor resultado foi entre todas (de acordo com as medidas retiradas, conseguiu melhor colocação na maioria das métricas) foi mostrado na Tabela 07.

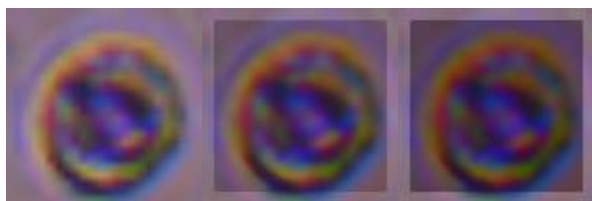
Na Tabela 07, estão os resultados das médias dos valores obtidos em cada tipo de mudança de porcentagem de *dropout*. É possível observar que com *dropout* melhora a variância do laplaciano. Entretanto, esse resultado não pode ser usado, uma vez que o resultado foi comprometido, pois ocorreu a formação de um “frame” (quadro) em volta da imagem, mostrado na Figura 48. Por isso, as mudanças feitas com *Dropout* foram excluídas e foi escolhida a que manteve o resultado sem a adição da camada de *Dropout*.

Tabela 7 - Resultado Final da Mudança do *Dropout*

	Camadas Dropout				
	0	0,1	0,2	0,3	0,4
PSNR	39,032	24,449	20,383	0,000	0,000
MSE (menor valor - melhor)	25,026	837,576	2106,885	0,000	0,000
SSIM	0,977	0,832	0,716	0,000	0,000
Var_Lap - L	23,709	73,537	197,820	0,000	0,000
Var_Lap - E/B	58,403	98,706	262,509	0,000	0,000
Var_Lap - M	16,020	97,061	286,317	0,000	0,000
Var_Lap - N	20,621	93,810	268,210	0,000	0,000

Fonte: autoria própria.

Figura 48 - Exemplo de *Frame* na Imagem de Saída com a mudança do *Dropout*



*Lendo da esquerda para direita, imagem normal, imagem com *dropout* de 10% e imagem com *dropout* de 20%.

Fonte: autoria própria.

A Figura 49 mostra o modelo que demonstrou o melhor resultado.

Figura 49 - Melhor Modelo com Mudanças do *Dropout*

```
def model():
    SRCNN = tf.keras.Sequential(name='SRCNN')
    SRCNN.add(tf.keras.layers.Conv2D(filters=8, kernel_size=(13, 13),
                                      padding='VALID',
                                      use_bias=True,
                                      input_shape=(None, None, 1),
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=16, kernel_size=(5, 5),
                                      padding='SAME',
                                      use_bias=True,
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=32, kernel_size=(5, 5),
                                      padding='SAME',
                                      use_bias=True,
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=64, kernel_size=(5, 5),
                                      padding='SAME',
                                      use_bias=True,
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=128, kernel_size=(5, 5),
                                      padding='SAME',
                                      use_bias=True,
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
```

Fonte: autoria própria.

Com esse resultado é possível observar que colocando as camadas de *dropout* obteve-se um efeito reverso e diminuiu de forma drástica a melhoria dos valores das métricas que foram, assim, retiradas e continuaram com o modelo anterior.

Com a mudança das funções de ativação, a que apresentou melhor resultado foi com a Função *Relu* (*Rectified Linear Unit*) mostrado na Tabela 8.

Tabela 8 - Resultado Final da Mudança das Funções de Ativação

	Função de Ativação				
	Elu	Relu	Selu	Sigmoid	TanH
PSNR	38,887	39,032	39,017	38,810	38,780
MSE (menor valor - melhor)	25,975	25,026	25,145	26,493	26,681
SSIM	0,977	0,977	0,977	0,976	0,976
Var_Lap - L	23,657	23,709	23,523	23,552	23,632
Var_Lap - E/B	57,331	58,403	54,689	52,677	52,148
Var_Lap - M	15,951	16,020	15,787	15,821	15,767
Var_Lap - N	19,858	20,621	20,191	19,747	20,108

Fonte: autoria própria.

Logo, entre todas essas mudanças, a estrutura que recebeu o melhor resultado foi a que está contido o 13 x 13 na primeira camada, como mostra a Tabela 9.

Tabela 9 - Resultado Final da Mudança de *kernel size* e *Padding*

	Kernal_Pad_Camada								
	1 Camada	2 Camada	3 Camada	4 Camada	5 Camada	2 Camada Segui	3 Camada Segui	4 Camada Segui	5 Camada Segui
PSNR	39,032	39,016	38,822	38,937	38,832	38,876	38,809	38,748	38,572
MSE (menor valor - melhor)	25,026	25,156	26,359	25,519	26,322	26,077	26,445	26,962	28,104
SSIM	0,977	0,978	0,977	0,977	0,977	0,977	0,977	0,976	0,975
Var_Lap - L	23,709	23,627	23,453	23,088	23,698	23,446	23,591	23,733	23,498
Var_Lap - E/B	58,403	56,865	54,625	52,856	52,797	52,856	52,414	50,036	48,678
Var_Lap - M	16,020	15,712	15,860	15,701	16,449	16,071	15,787	16,048	15,774
Var_Lap - N	20,621	20,292	20,007	19,987	20,586	20,110	20,044	19,977	19,290

Fonte: autoria própria.

Em seguida, realizou-se a mudança do *Bias*, entre *True* e *False*, sendo que o melhor resultado foi com o parâmetro ajustado em *True*, como mostra a Tabela 10.

Tabela 10 - Resultado Final da Mudança do *Bias*

	Bias	
	True	False
PSNR	39,032	38,755
MSE (menor valor - melhor)	25,026	26,869
SSIM	0,977	0,976
Var_Lap - L	23,709	23,030
Var_Lap - E/B	58,403	50,678
Var_Lap - M	16,020	16,024
Var_Lap - N	20,621	19,956

Fonte: autoria própria.

Logo após, foi modificado o *Kernel* Iniciador, entre *Glorot Uniform*, *Random Uniform* e *He Uniform*, o qual obteve melhor resultado foi com *Glorot Uniform*, como mostra a Tabela 11.

Tabela 11 - Resultado Final da Mudança do *Kernel Initializer*

	Kernel Iniciador		
	GU	RU	HU
PSNR	39,032	38,765	38,788
MSE (menor valor - melhor)	25,026	26,804	26,653
SSIM	0,977	0,976	0,976
Var_Lap - L	23,709	23,250	23,422
Var_Lap - E/B	58,403	49,989	51,310
Var_Lap - M	16,020	16,202	16,121
Var_Lap - N	20,621	19,822	19,612

Fonte: autoria própria.

Já na mudança do otimizador (entre Adam, SGD e Ftrl), houve a montagem de quadros nas imagens com os otimizadores SGD e Ftrl, sendo possível somente utilizar o Adam para seguir para a próxima etapa, como mostrado na Tabela 12.

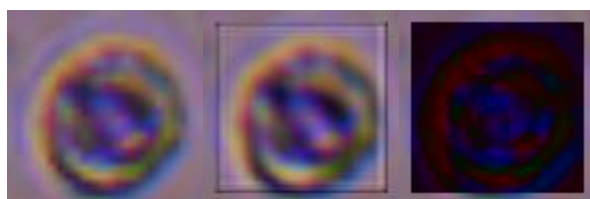
Tabela 12 - Resultado Final da Mudança do Otimizador

	Otimizador		
	Adam	SGD	Ftrl
PSNR	39,032	17,262	12,732
MSE (menor valor - melhor)	25,026	5870,299	11701,727
SSIM	0,977	0,387	0,259
Var_Lap - L	23,709	1002,582	1182,262
Var_Lap - E/B	58,403	1095,432	1384,441
Var_Lap - M	16,020	1314,733	1634,148
Var_Lap - N	20,621	1151,405	1321,491

Fonte: autoria própria.

Na Figura 50, é possível observar os diferentes tipos de *frames* causados pelos dois otimizadores, do SGD e Ftrl, respectivamente.

Figura 50 - Exemplo de Frame na Imagem de Saída com Otimizador SGD



*Lendo da esquerda para direita, imagem normal, imagem com Otimizador SGD e imagem com Otimizador Ftrl.

Fonte: autoria própria.

Na última modificação, do *Learning Rate* (LR), usando valores com casas decimais diferentes (0,00005; 0,0003 e 0,005) para observar qual a diferença ocorre com os valores alterados, houve novamente a criação do quadro envolta da imagem com o parâmetro ajustado em 0,005. Assim, o “0,0003” alcançou melhor resultado, como mostra a Tabela 13.

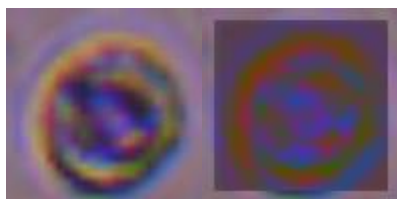
Na Figura 51, é possível observar o diferente frame causado pelo LR com “0,005”.

Assim, após feitos todos os testes anteriores, foi alcançada a melhor estrutura de SRCNN, demonstrado na Figura 52.

Tabela 13 - Resultado Final da Mudança do Taxa de Aprendizado

	Learning Rate		
	0,000	0,000	0,005
PSNR	38,729	39,032	18,390
MSE (menor valor - melhor)	27,072	25,026	5708,864
SSIM	0,975	0,977	0,464
Var_Lap - L	22,961	23,709	523,848
Var_Lap - E/B	49,600	58,403	663,477
Var_Lap - M	15,369	16,020	767,355
Var_Lap - N	20,311	20,621	608,757

Fonte: autoria própria.

Figura 51 - Exemplo de Frame na Imagem de Saída com diferentes *Learning Rates*

*Lendo da esquerda para direita, imagem normal, imagem com LR = 0,005.

Fonte: autoria própria.

Figura 52 - Modelo Final “Vencedor”

```
def model():
    SRCNN = tf.keras.Sequential(name='SRCNN')
    SRCNN.add(tf.keras.layers.Conv2D(filters=8, kernel_size=(13, 13),
                                      padding='VALID',
                                      use_bias=True,
                                      input_shape=(None, None, 1),
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=16, kernel_size=(5, 5),
                                      padding='SAME',
                                      use_bias=True,
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=32, kernel_size=(5, 5),
                                      padding='SAME',
                                      use_bias=True,
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=64, kernel_size=(5, 5),
                                      padding='SAME',
                                      use_bias=True,
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))
    SRCNN.add(tf.keras.layers.Conv2D(filters=128, kernel_size=(5, 5),
                                      padding='SAME',
                                      use_bias=True,
                                      kernel_initializer='glorot_uniform',
                                      activation='relu'))

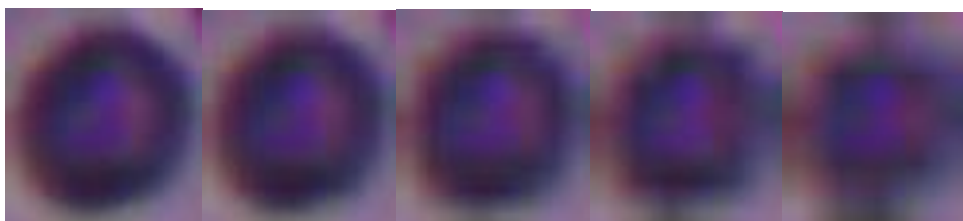
    # Optimizer/Otimizador = encontrar o ponto minino de uma função (no caso a
    # Os pesos de uma REde Neural são aprendidos com os otimizadores
    # Exemplos: Descida do Gradiente - atualiza o peso com a inclinação para m
    # Taxa de aprendizagem
    optimizer = tf.keras.optimizers.Adam(learning_rate=0.0003)
```

Fonte: autoria própria.

Com a estrutura final escolhida, foi possível recriar todas as imagens com SR que estão nas pastas de Treinamento, Validação e Testes para serem utilizadas na CNN de classificação.

No *Jupyter*, o primeiro teste das métricas de SR foi com fator/escala diferentes com o melhor modelo descoberto, com valores 2, 4, 6, 8 e 10, para descobrir se a SRCNN tem um "limite" máximo de melhoramento, que depois de uma certa escala, não consegue melhorar muito mais. Assim, foram coletados pontos para fazer a linearidade, criando um gráfico para observar até onde vale a pena fazer a SR, mostrados nas Figuras 55, 56 e 57. Nesse teste, foi utilizada uma imagem de um linfócito aleatório, sendo empregada a mesma imagem em todos os testes. Na Figura 53, estão as imagens de entrada com a quantidade de escala que foram diminuídas e na Figura 54, estão as imagens de saída das imagens de entrada, citadas anteriormente.

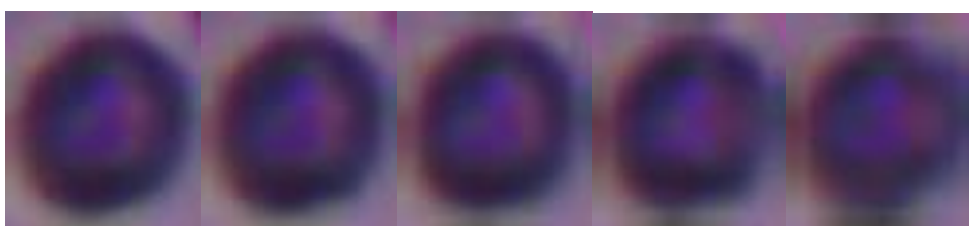
Figura 53 - Imagens de Entrada para Teste com diferentes Escalas



*Lendo da esquerda para direita, imagem de entrada com escala diminuída 2x da original, imagem de entrada com escala diminuída 4x da original, imagem de entrada com escala diminuída 6x da original, imagem de entrada com escala diminuída 8x da original e imagem de entrada com escala diminuída 10x da original,

Fonte: autoria própria.

Figura 54 - Imagens de Saída para Teste com diferentes Escalas

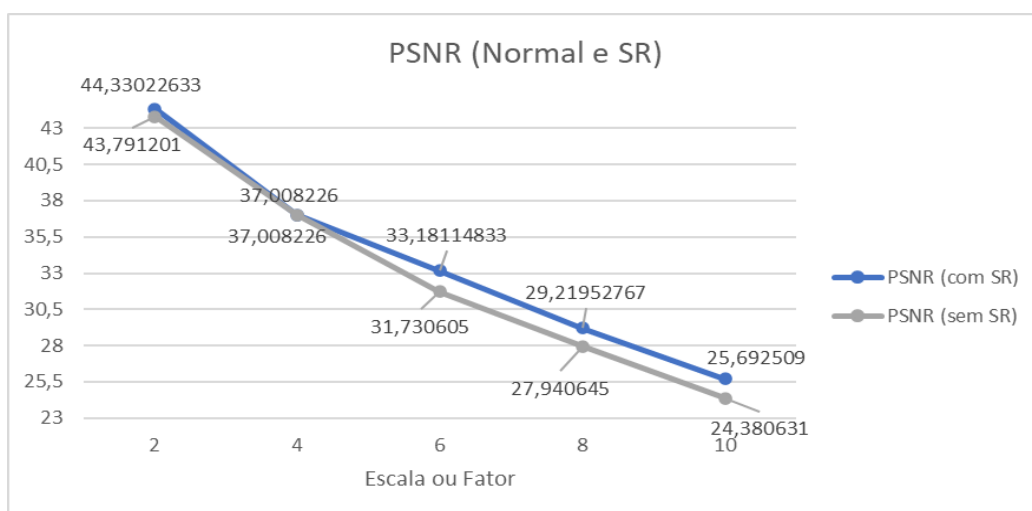


*Lendo da esquerda para direita, imagem de saída da escala que diminuiu 2x, imagem de saída da escala que diminuiu 4x, imagem de saída da escala que diminuiu 6x, imagem de saída da escala que diminuiu 8x e imagem de saída da escala que diminuiu 10x,

Fonte: autoria própria.

Como é possível observar, na Figura 55 é representada a métrica PSNR tanto para as imagens com Super Resolução (SR) e sem (normal), saindo da escala 2 até alcançar o valor 10. Verifique-se que, com a SR, houve uma queda do valor de 44,33022633 para 25,692509, diminuindo em aproximadamente 57% a melhora da imagem (da escala inicial 2 até a final 10). Já das imagens “normais” houve uma queda de 43,791201 para 24,380631 representando uma baixa de aproximadamente 55%, percebendo-se a similaridade nas quedas, mantendo uma certa curva.

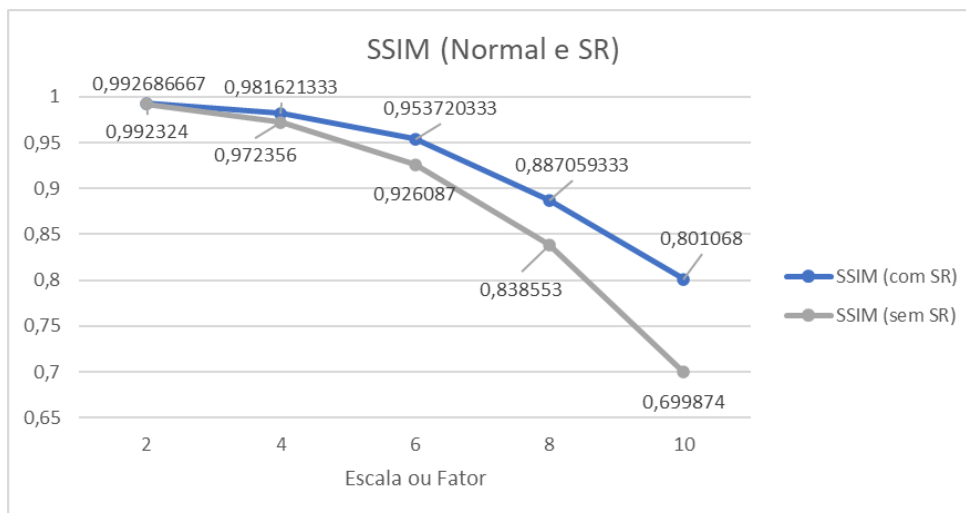
Figura 55 - Super Resolução - Gráficos Resultados PSNR



Fonte: autoria própria.

Agora, na Figura 56, comparando outra métrica SSIM, novamente comparando com SR e normal, novamente tendo uma queda quando a escala vai de 2 para 10. Entretanto, enquanto na escala de fator 2 mantém-se um valor extremamente próximo, com o passar das escalas, alcançando a última, ocorre uma diferença bem distinta entre os valores das imagens normais e de SR. Provando que as imagens com SR, possuem mais similaridade na estrutura que as imagens que somente sofrem redimensionamento.

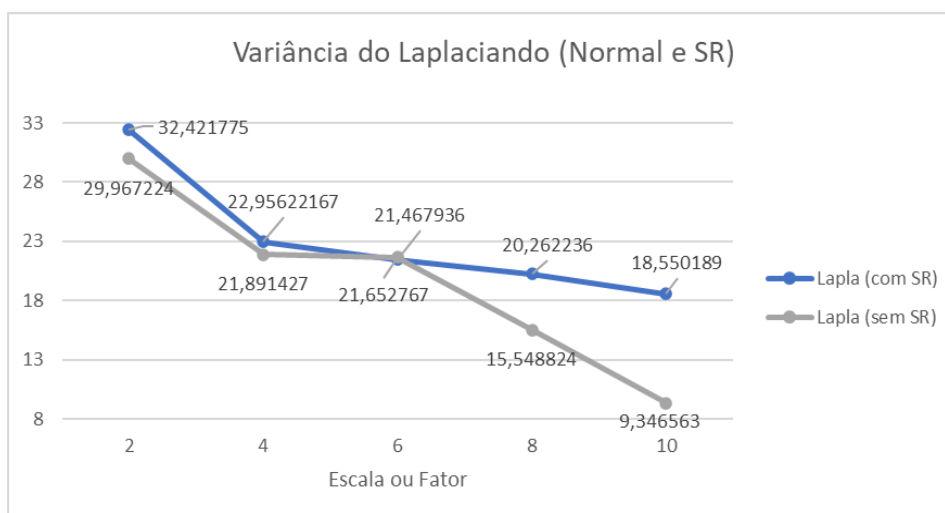
Figura 56 - Super Resolução - Gráficos Resultados SSIM



Fonte: autoria própria.

Já na Figura 57, comparando a última métrica, que é Variância do Laplaciano, novamente comparando entre SR e normal, o gráfico apresenta um comportamento diferente do que foi visto anteriormente, pois quando alcança “Escala ou Fator” de valor 6, ocorre a inversão em que a imagem sem SR possui um “melhor foco” comparado com a imagem criada pelo algoritmo de SR. Nas outras escalas, a imagem de SR apresenta um melhor foco comparado com a imagem normal de entrada. Assim, no geral, as imagens que foram criadas por SR apresentam melhor foco em comparação com as imagens que somente fazem o redimensionamento.

Figura 57 - Super Resolução - Gráficos Resultados Variância do Laplaciano



Fonte: autoria própria.

Os testes dos tamanhos diferentes de imagens (70x70, 85x85 e 100x100) alcançaram, em todas as métricas, valores entre aproximadamente 95% a 96,3% na CNN e 95,7% a 99,7%, utilizando *dataset* que contém 71.010 imagens (49.707 para treinamento e 21.303 para teste, como explicado anteriormente), conseguindo atingir os valores propostos na seção Hipótese. As imagens de tamanhos menores obtiveram menores porcentagem e as maiores obtiveram as melhores. Porém, não houve uma variação alta. Assim, foram escolhidas as imagens de 85 x 85, por causa do tempo de rodar o código para não ocorrer o *TimeOut*, uma vez que estava rodando no Google Colab.

Com o tamanho escolhido e quantidade das imagens, tanto para a pasta de treinamento e de teste feitas, realizou-se o teste somente com imagens criadas a partir da SR, tanto para a CNN quanto para a SVM, para observar a comparação de desempenho com o *dataset* de imagens “normais” e um *dataset* de imagens somente de SR.

Para o classificador da CNN, que obteve um desempenho próximo de 96% nas métricas de classificação com as imagens normais e aproximadamente com 91% com as imagens de SR, os resultados são mostrados na Figura 58.

Figura 58 - Resultados do Teste da CNN com Imagens Originais e com Imagens de SR

Accuracy: 0.9632446134347274	Accuracy: 0.9117025771018167
Precision: 0.9633362390042377	Precision: 0.9142137212166903
Sensitivity_recall: 0.9632446134347274	Sensitivity_recall: 0.9117025771018167
Specificity: 0.9632446134347274	Specificity: 0.9117025771018167
F1_score: 0.9630934580085347	F1_score: 0.911015354650442

Fonte: autoria própria.

Já para a SVM, obteve-se um desempenho próximo de 98% com as imagens normais e 89% com as imagens de SR, como mostra a Figura 59.

Figura 59 - Resultados do Teste da SVM com Imagens Originais e com Imagens de SR

'Accuracy': 0.9868093695723607,	'Accuracy': 0.8848518987935972,
'Precision': 0.987193569476191,	'Precision': 0.9107183781547782,
'Sensitivity_recall': 0.9868093695723607,	'Sensitivity_recall': 0.8848518987935972,
'Specificity': 0.9868093695723607,	'Specificity': 0.8848518987935972,
'F1_score': 0.9867261564135332}	'F1_score': 0.8787688164734635}

Fonte: autoria própria.

Assim, foi possível observar que houve uma queda no desempenho do classificador para as imagens. Ressalte-se que na seção dos métodos já foram apontadas outras duas alternativas para melhorar o desempenho, a de mudança do *dataset*, como explicado anteriormente. Entretanto, antes foi testado o algoritmo com somente imagens “normais” e com 100% das imagens de SR.

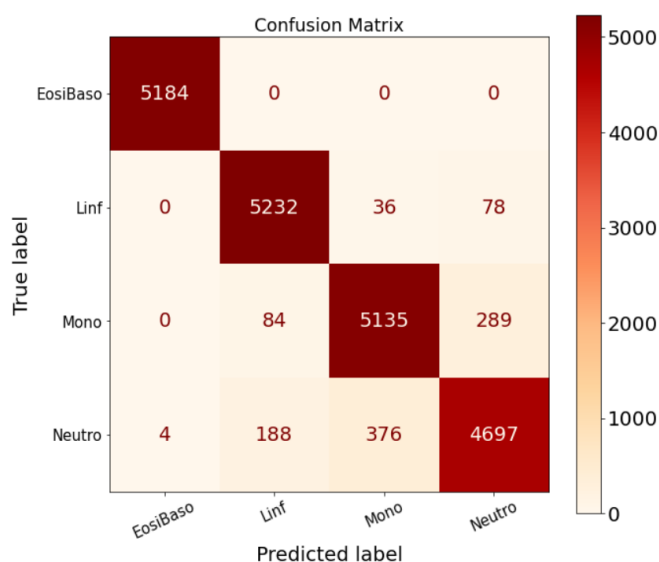
A primeira tentativa foi no *dataset* 100% original, sem nenhuma imagem com SR (observado na Figura 60), tem-se o gráfico da acurácia e também de perda, tanto no treinamento quanto na validação, com seus valores com o passar de cada época de treinamento. Já na Figura 61, tem-se a matriz de confusão com o resultado de cada classe.

Figura 60 - Gráficos de Aprendizado e Valores por Época – 100% das Imagens Originais



Fonte: autoria própria.

Figura 61 - Matriz de Confusão – 100% das Imagens Originais



Fonte: autoria própria.

Também foi analisado o “*Classification Report*”, ou Relatório de Classificação, o qual mostra as principais métricas de classificação de cada classe, como mostrado na parte esquerda da Figura 62. Já na parte direita, estão as métricas (explicadas anteriormente), entretanto fazendo os cálculos considerando todas as classes.

Figura 62 - Relatório de Classificação e Métricas de Classificação – 100% das Imagens Originais

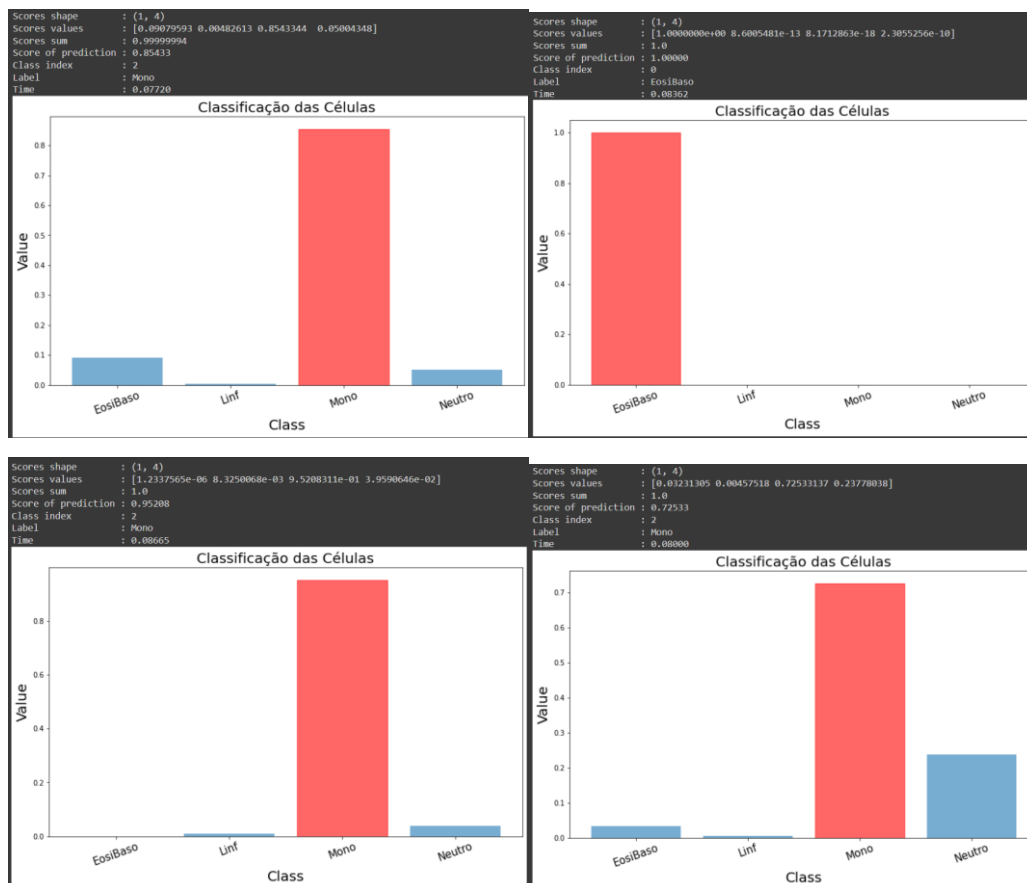
	precision	recall	f1-score	support	
0	1.00	1.00	1.00	5184	Accuracy: 0.950476458714735 Precision: 0.9502953096449528 Sensitivity_recall: 0.950476458714735 Specificity: 0.950476458714735 F1_score: 0.9502463738300514
1	0.95	0.98	0.96	5346	
2	0.93	0.93	0.93	5508	
3	0.93	0.89	0.91	5265	
accuracy			0.95	21303	
macro avg	0.95	0.95	0.95	21303	
weighted avg	0.95	0.95	0.95	21303	

Fonte: autoria própria.

Dessa forma, o algoritmo 100% com imagens originais, classificou as imagens de LR da seguinte forma (observando a Figura 63 da esquerda para direita de cima para baixo): o linfócito foi classificado em monócito com aproximadamente 85%, o eosinófilo foi classificado corretamente com 100% de certeza, como também o monócito com aproximadamente 95%, já o neutrófilo foi classificado como monócito com 72%.

Já com as imagens de SR, o algoritmo classificou do seguinte modo (observando a Figura 64 da esquerda para direita de cima para baixo): o linfócito foi classificado em eosinófilo com aproximadamente 82%, o eosinófilo foi classificado corretamente com 100% de certeza, como também o monócito com aproximadamente 93% e também o neutrófilo foi classificado com 61%.

Figura 63 - Classificação das Imagens de LR – 100% das Imagens Originais

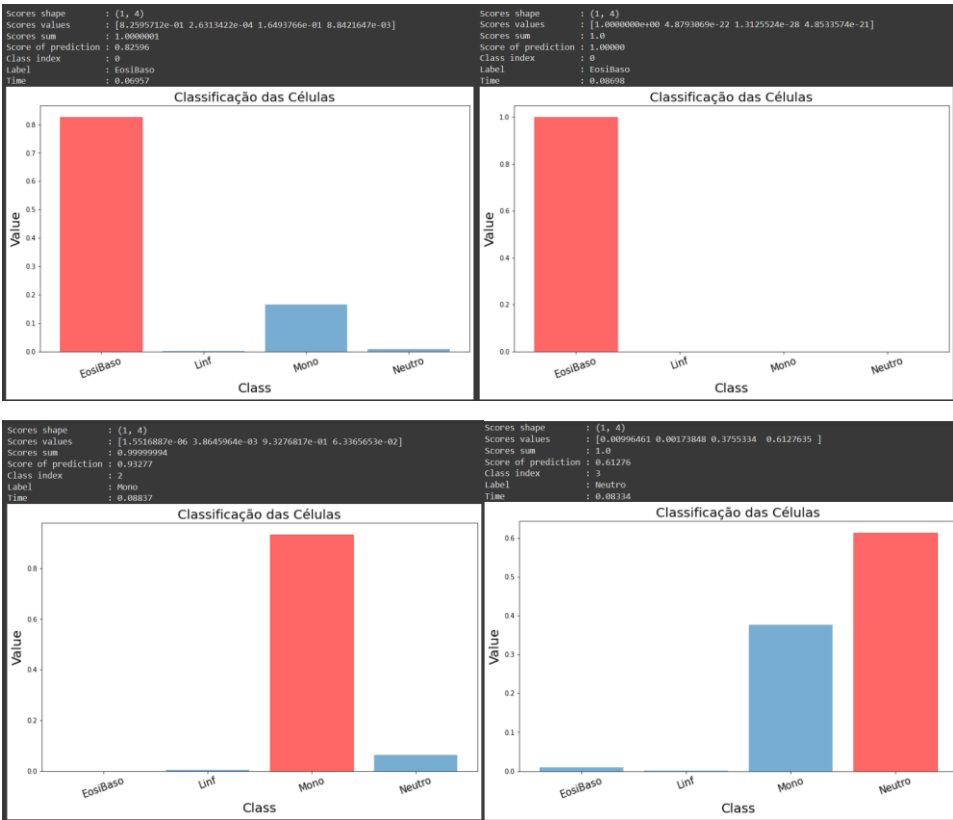


Fonte: autoria própria.

Desse modo, é possível observar que a imagem de SR, mais especificamente a do neutrófilo, teve a possibilidade de ser classificada corretamente, em contraste com a de LR que obteve a classificação errada. Provando, assim, que usar imagens com qualidades inferiores no algoritmo de SR para obter melhores resultados, ou seja, imagens com qualidades superiores podem fazer uma grande diferença na classificação das mesmas.

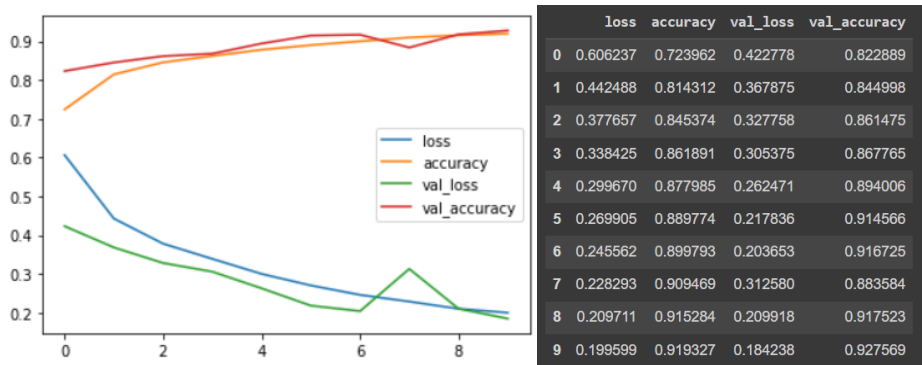
A segunda tentativa foi no *dataset* 100% com imagens de SR, é possível observar na Figura 65, tem-se o gráfico da acurácia e também de perda, tanto no treinamento quanto na validação, com seus valores com o passar de cada época de treinamento. Já na Figura 66, tem-se a matriz de confusão com o resultado de cada classe.

Figura 64 - Classificação das Imagens de SR – 100% das Imagens Originais



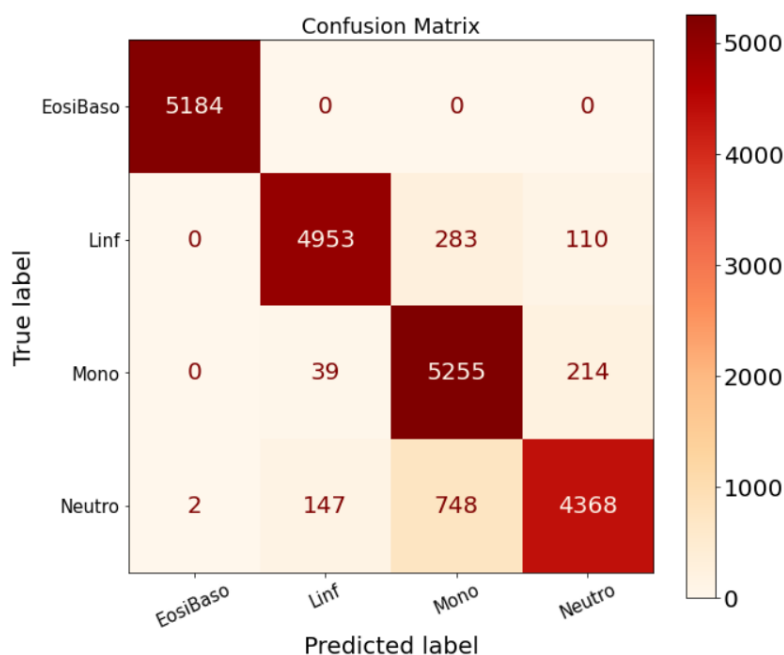
Fonte: autoria própria.

Figura 65 - Gráficos de Aprendizado e Valores por Época – 100% das Imagens de SR



Fonte: autoria própria.

Figura 66 - Matriz de Confusão – 100% das Imagens de SR



Fonte: autoria própria.

Também foi analisado o “*Classification Report*”, ou Relatório de Classificação, o qual mostra as principais métricas de classificação de cada classe, como mostrado na parte esquerda da Figura 67. Já na parte direita, estão as métricas (explicadas anteriormente), entretanto, fazendo os cálculos considerando todas as classes.

Figura 67 - Relatório de Classificação e Métricas de Classificação – 100% das Imagens de SR

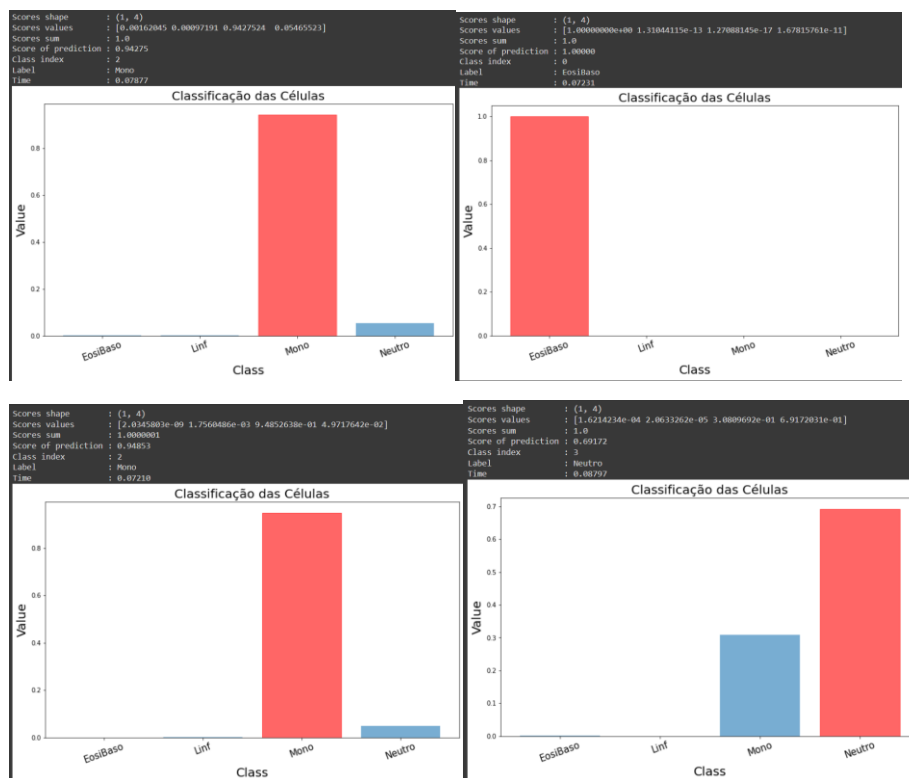
	precision	recall	f1-score	support	
0	1.00	1.00	1.00	5184	Accuracy: 0.9275688870112191
1	0.96	0.93	0.94	5346	Precision: 0.9313497985693907
2	0.84	0.95	0.89	5508	Sensitivity_recall: 0.9275688870112191
3	0.93	0.83	0.88	5265	Specificity: 0.9275688870112191
accuracy			0.93	21303	F1_score: 0.9276393949209094
macro avg	0.93	0.93	0.93	21303	
weighted avg	0.93	0.93	0.93	21303	

Fonte: autoria própria.

Após a obter e observar os valores das métricas, foram apresentados os testes das imagens que são de LR e de SR, como demonstram as Figuras 68 e 69, já mostradas anteriormente. Desse modo, com imagens originais, o algoritmo classificou as imagens de LR da seguinte forma (observando a Figura 68 da esquerda para direita de cima para baixo): o linfócito foi classificado em monócito com aproximadamente 94%, o eosinófilo foi classificado corretamente com 100% de certeza, como também

o monócito com aproximadamente 94% e também o neutrófilo foi classificado corretamente com 69%.

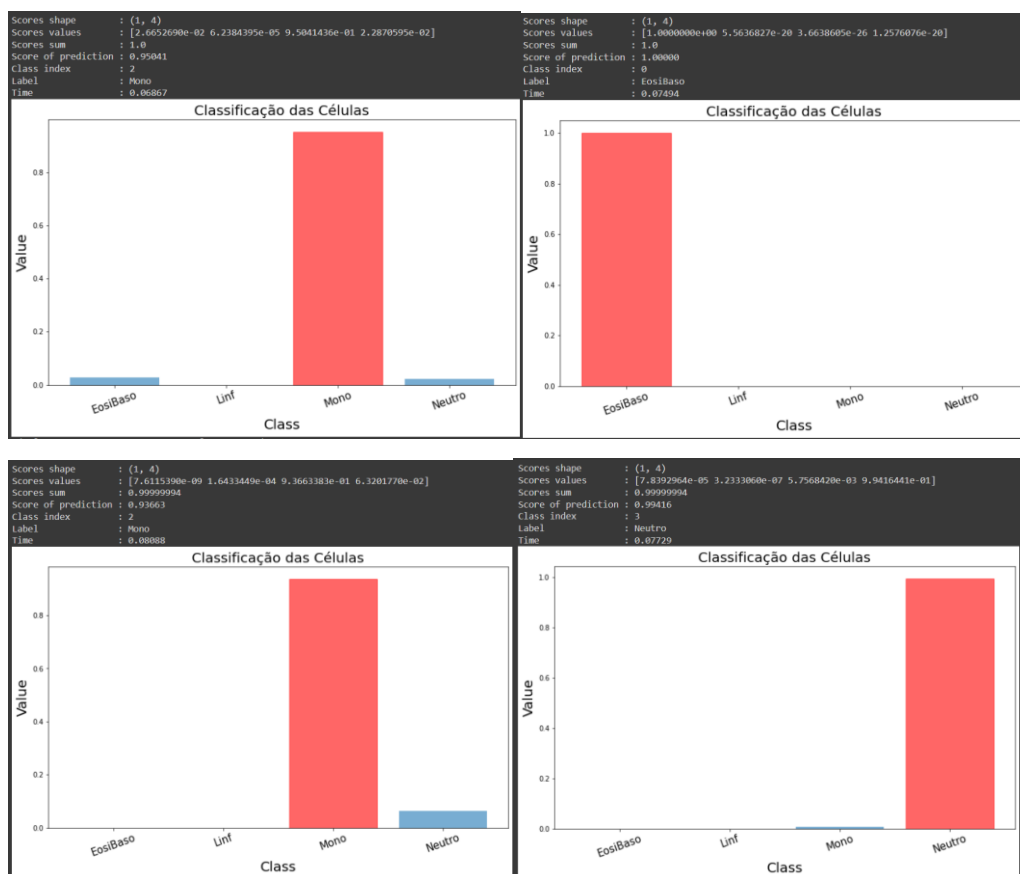
Figura 68 - Classificação das Imagens de LR – 100% das Imagens de SR



Fonte: autoria própria.

Já com as imagens de SR, o algoritmo classificou do seguinte modo (observando a Figura 69 da esquerda para direita de cima para baixo): o linfócito foi classificado novamente em monócito com aproximadamente 95%, o eosinófilo foi classificado corretamente com 100% de certeza, como também o monócito com aproximadamente 93% e também o neutrófilo foi classificado com 99%.

Figura 69 - Classificação das Imagens de SR – 100% das Imagens de SR

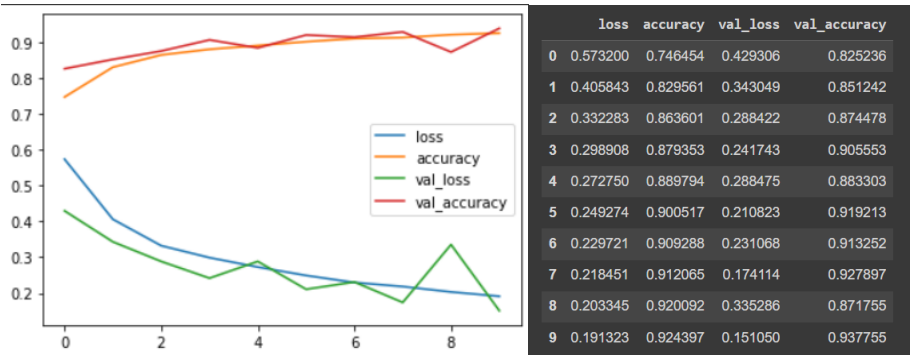


Fonte: autoria própria.

Assim, observando os resultados, focando na imagem do neutrófilo, tanto de LR e de SR, ambos obtiveram as classificações corretas. Entretanto, a imagem com SR atingiu aproximadamente 99% enquanto a imagem de LR teve 69%. Desse modo, com o treinamento com imagens de SR, conseguiu-se não só classificar corretamente a imagem de SR, como também a imagem de LR, demonstrando que treinar com imagens também feitas por SR pode ser um caminho de interesse para obter melhores resultados em imagens com qualidades inferiores, mesmo obtendo resultados menores nas métricas, comparado com o *dataset* 100% de imagens originais.

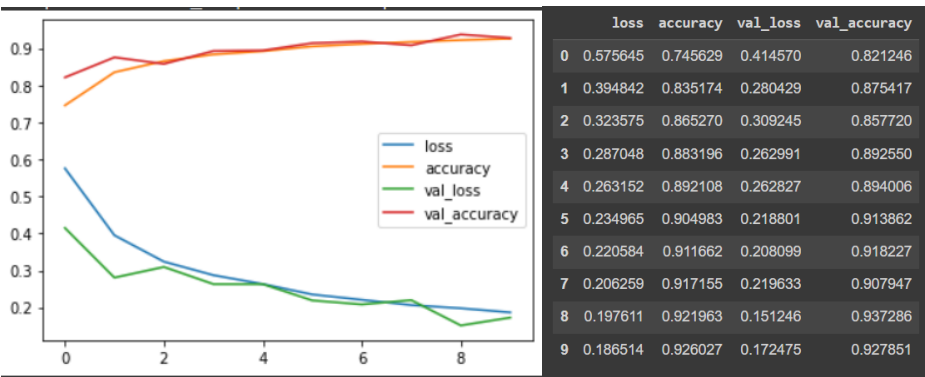
É possível observar nas Figura 70, 71 e 72 os gráficos da acurácia e também de perda do primeiro, segundo e terceiro testes de substituição, tanto no treinamento quanto na validação, com seus valores com o passar de cada época de treinamento. Já na Figura 73, observam-se as matrizes de confusão com o resultado de cada classe, observando de cima para baixo: primeiro, segundo e terceiro testes.

Figura 70 - Gráficos de Aprendizado e Valores por Época – Primeira mudança de Substituição



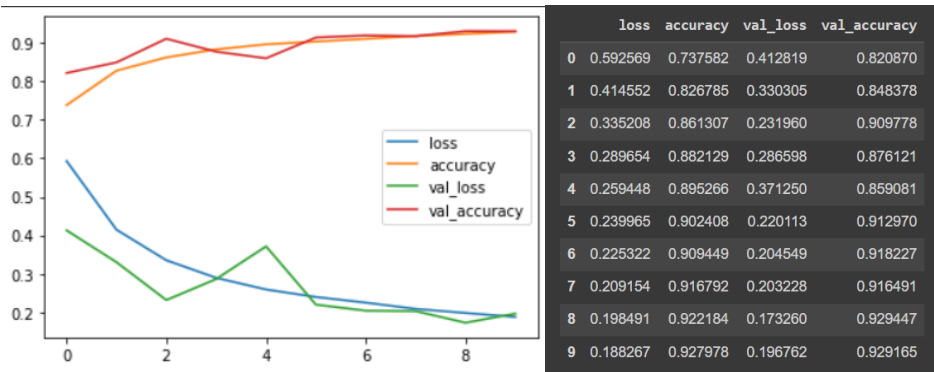
Fonte: autoria própria.

Figura 71 - Gráficos de Aprendizado e Valores por Época – Segunda mudança de Substituição



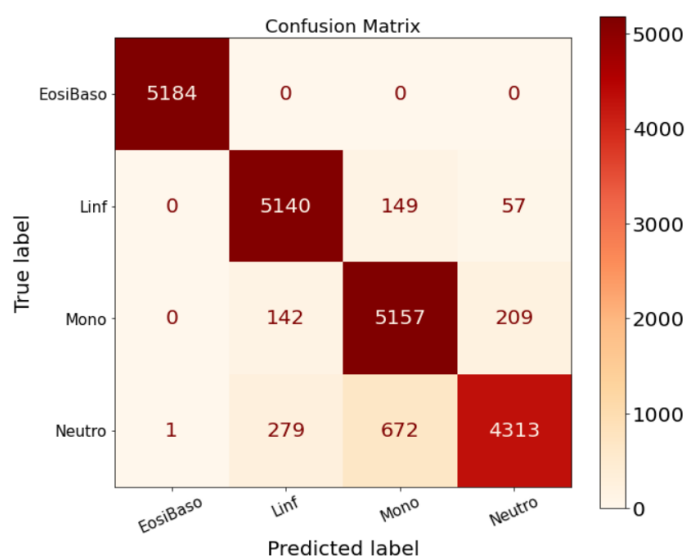
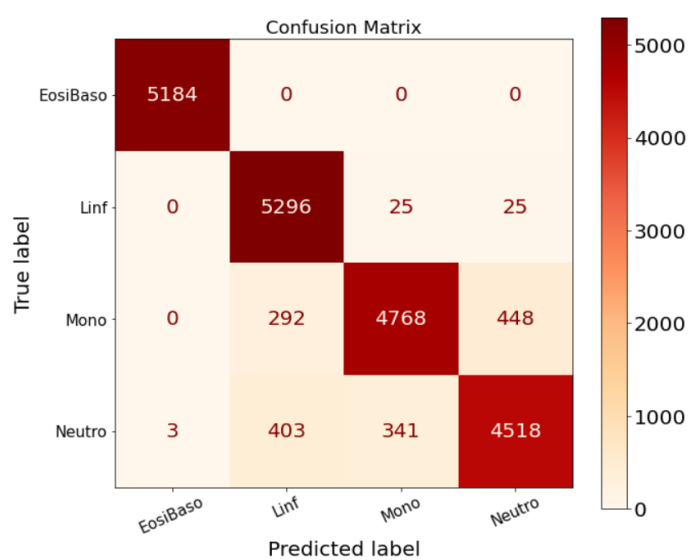
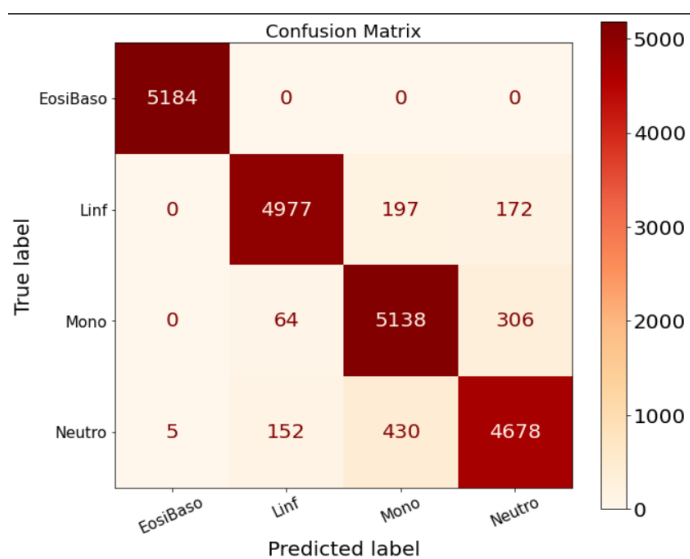
Fonte: autoria própria.

Figura 72 - Gráficos de Aprendizado e Valores por Época – Terceira mudança de Substituição



Fonte: autoria própria.

Figura 73 - Matriz de Confusão – Primeira, Segunda e Terceira mudanças de Substituição



Fonte: autoria própria.

Também foi analisado o Relatório de Classificação (Figuras 74, 75 e 76), já explicado anteriormente, como mostrado na parte esquerda das figuras. Já na parte direita, estão as métricas (explicadas anteriormente), entretanto fazendo os cálculos considerando todas as classes.

Figura 74 - Relatório de Classificação e Métricas de Classificação – Primeira mudança de Substituição

	precision	recall	f1-score	support	
0	1.00	1.00	1.00	5184	Accuracy: 0.9377552457400367
1	0.96	0.93	0.94	5346	Precision: 0.9382944657443314
2	0.89	0.93	0.91	5508	Sensitivity_recall: 0.9377552457400367
3	0.91	0.89	0.90	5265	Specificity: 0.9377552457400367
accuracy			0.94	21303	F1_score: 0.9378279957031587
macro avg	0.94	0.94	0.94	21303	
weighted avg	0.94	0.94	0.94	21303	

Fonte: autoria própria.

Figura 75 - Relatório de Classificação e Métricas de Classificação – Segunda mudança de Substituição

	precision	recall	f1-score	support	
0	1.00	1.00	1.00	5184	Accuracy: 0.9278505374829836
1	0.88	0.99	0.93	5346	Precision: 0.9288925071446975
2	0.93	0.87	0.90	5508	Sensitivity_recall: 0.9278505374829836
3	0.91	0.86	0.88	5265	Specificity: 0.9278505374829836
accuracy			0.93	21303	F1_score: 0.9271680938414519
macro avg	0.93	0.93	0.93	21303	
weighted avg	0.93	0.93	0.93	21303	

Fonte: autoria própria.

Figura 76 - Relatório de Classificação e Métricas de Classificação – Terceira mudança de Substituição

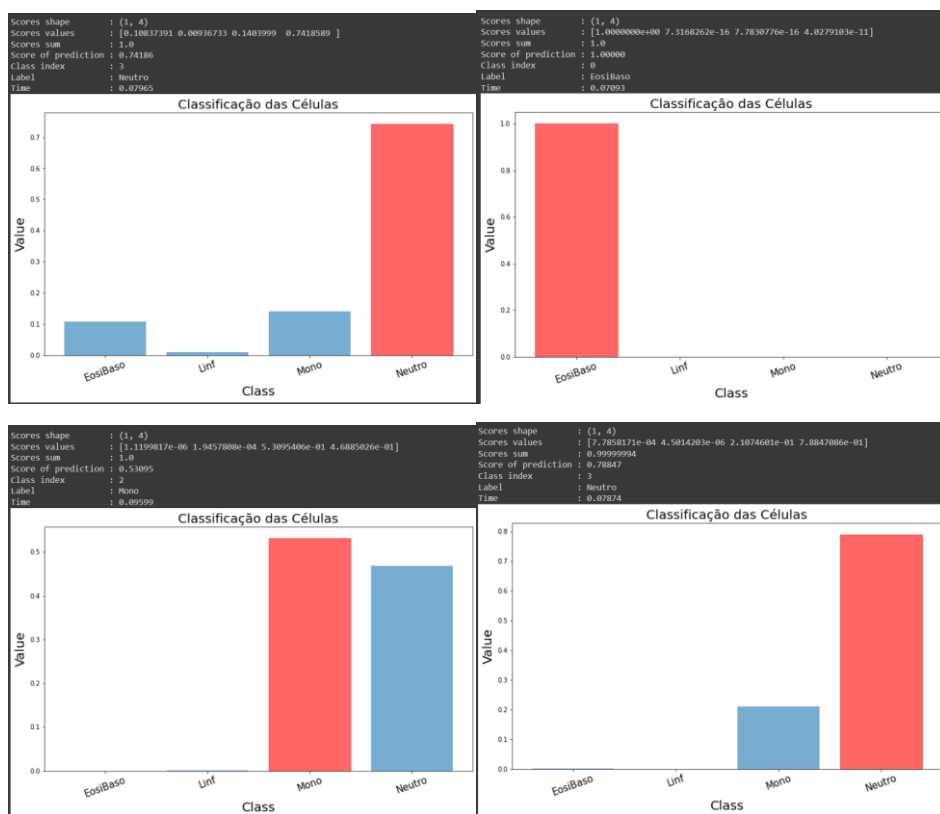
	precision	recall	f1-score	support	
0	1.00	1.00	1.00	5184	Accuracy: 0.9291649063512182
1	0.92	0.96	0.94	5346	Precision: 0.931088328553317
2	0.86	0.94	0.90	5508	Sensitivity_recall: 0.9291649063512182
3	0.94	0.82	0.88	5265	Specificity: 0.9291649063512182
accuracy			0.93	21303	F1_score: 0.9285884193030846
macro avg	0.93	0.93	0.93	21303	
weighted avg	0.93	0.93	0.93	21303	

Fonte: autoria própria.

Após observar os valores das métricas, foram feitos os testes das imagens que são de LR e de SR, como apresentado nas Figuras 53 e 54, já mostradas anteriormente.

A primeira mudança (primeiro teste) classificou as imagens de LR dessa forma (observando a Figura 77 da esquerda para direita de cima para baixo): o linfócito foi classificado em neutrófilo com aproximadamente 74%, o eosinófilo foi classificado corretamente com 100% de certeza, como também o monócito com aproximadamente 53% e também o neutrófilo foi classificado corretamente com 78%.

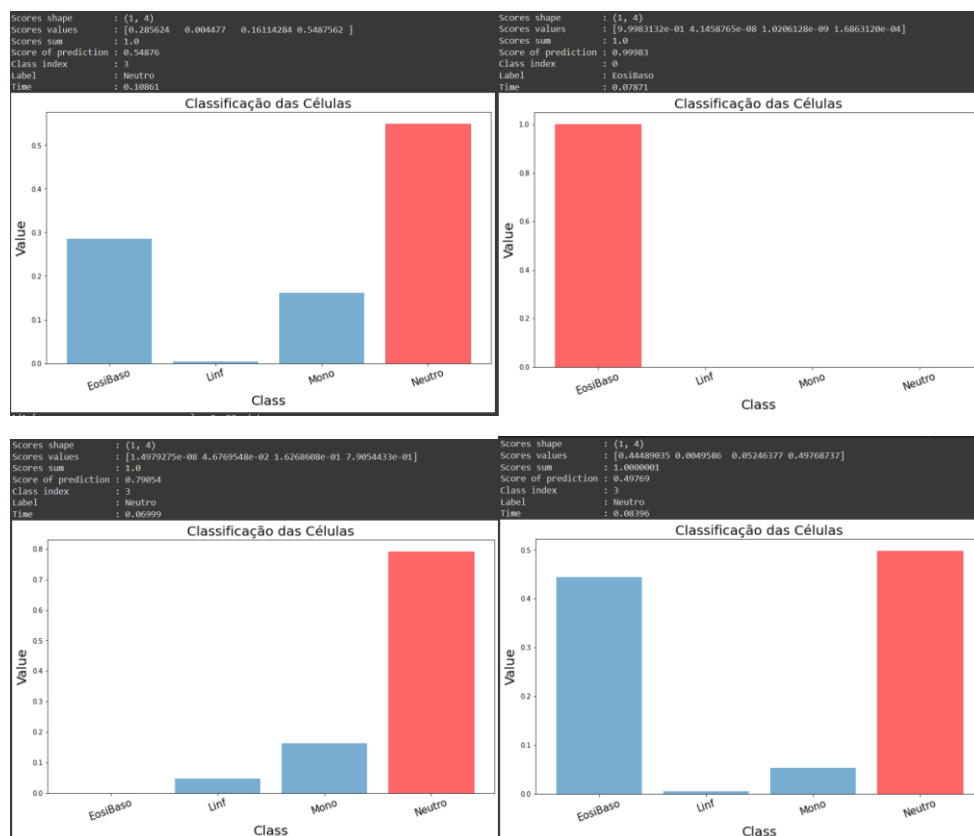
Figura 77 - Classificação das Imagens de LR – Primeira mudança de Substituição



Fonte: autoria própria.

A segunda mudança (segundo teste) classificou as imagens de LR da seguinte maneira (novamente observando a Figura 78 da esquerda para direita de cima para baixo): o linfócito foi classificado em neutrófilo com aproximadamente 54%, o eosinófilo foi classificado corretamente com 99% de certeza, como também o neutrófilo foi classificado corretamente com 49%, já o monócito foi classificado como neutrófilo com aproximadamente 79%.

Figura 78 - Classificação das Imagens de LR – Segunda mudança de Substituição

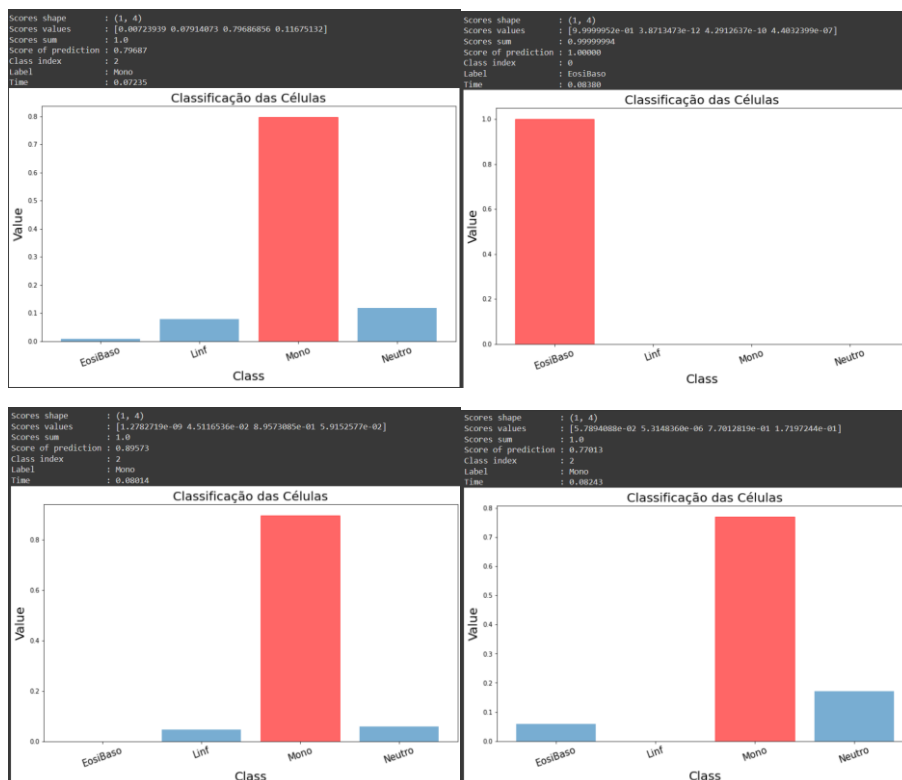


Fonte: autoria própria.

A terceira mudança (terceiro teste) classificou as imagens de LR da seguinte maneira (novamente observando a Figura 79 da esquerda para direita de cima para baixo): o linfócito foi classificado em monócito com aproximadamente 79%, o eosinófilo foi classificado corretamente com 100% de certeza, como também o monócito foi classificado corretamente com 89%, já o neutrófilo foi classificado como monócito com aproximadamente 77%.

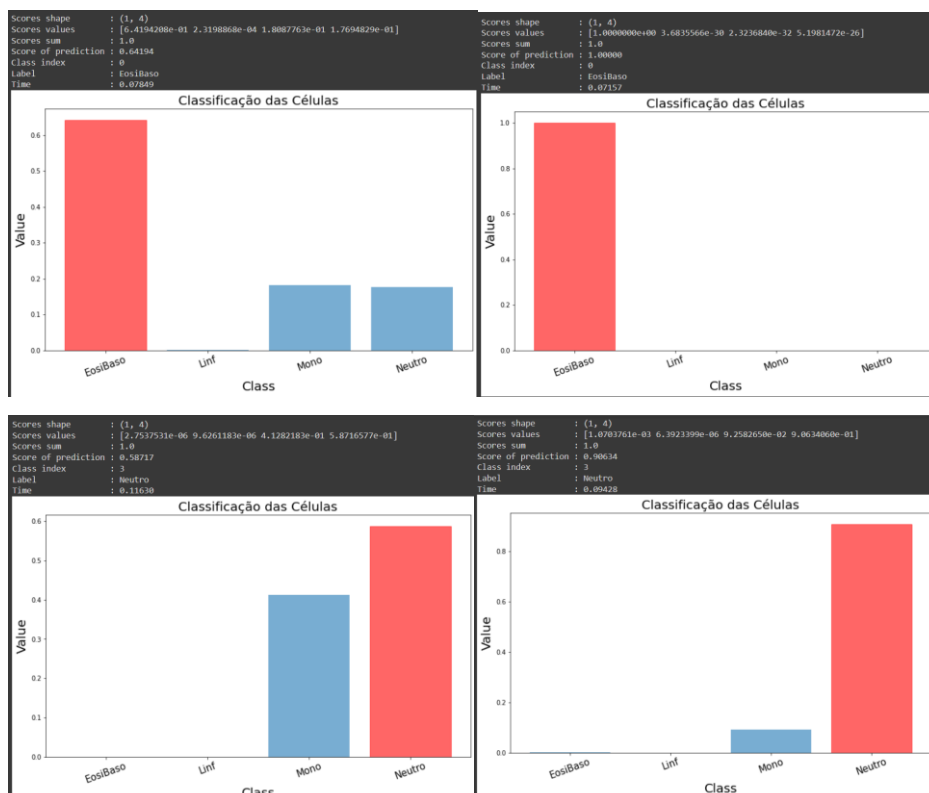
Assim, com as imagens de SR, o primeiro teste classificou do seguinte modo (observando a Figura 80 da esquerda para direita de cima para baixo): o linfócito foi classificado como eosinófilo com 64%, como também o monócito foi classificado erroneamente como neutrófilo com 58%, o eosinófilo foi classificado corretamente com 100% de acerto e também o neutrófilo foi classificado com 90% de acerto.

Figura 79 - Classificação das Imagens de LR – Terceira mudança de Substituição



Fonte: autoria própria.

Figura 80 - Classificação das Imagens de SR – Primeira mudança de Substituição

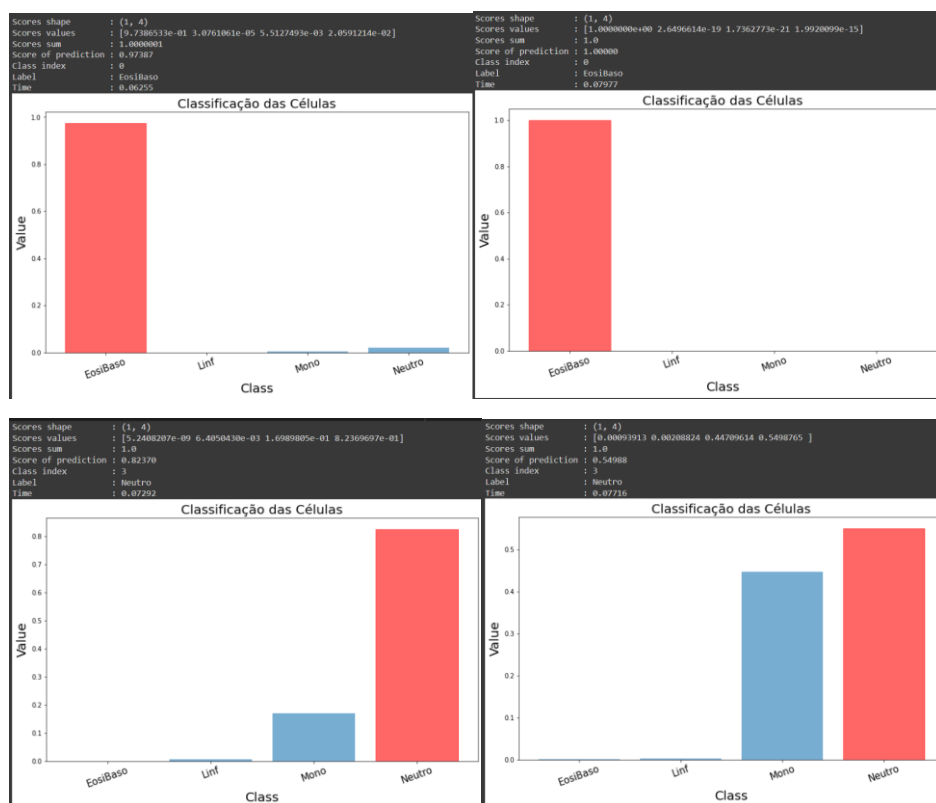


Fonte: autoria própria.

Novamente, com as imagens de SR, entretanto agora com o segundo teste, classificou do seguinte modo (observando a Figura 81 da esquerda para direita de cima para baixo): o linfócito foi classificado como eosinófilo com 97%, como também o monócito foi classificado erroneamente como neutrófilo com 82%, o eosinófilo foi classificado corretamente com 100% de acerto e também o neutrófilo foi classificado com 54% de acerto.

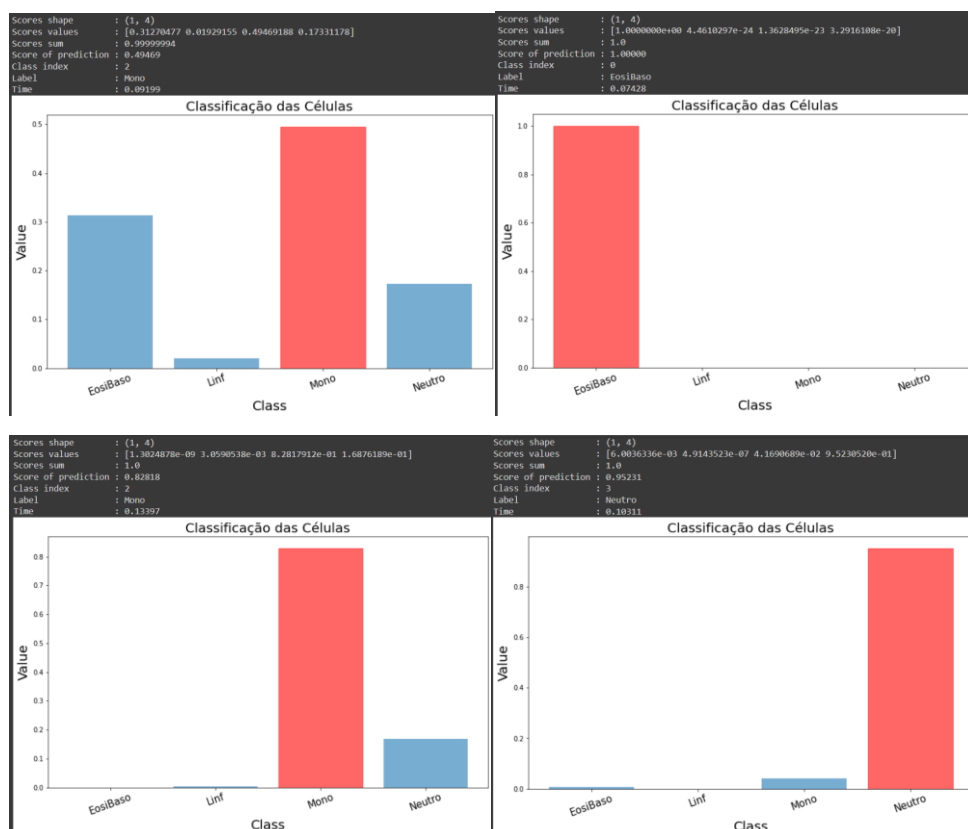
O terceiro algoritmo, com as imagens de SR classificou do seguinte modo (observando a Figura 82 da esquerda para direita de cima para baixo): o linfócito foi classificado como monócito com 49%, já o monócito, eosinófilo e neutrófilo foram classificados corretamente com 82%, 100% e 95%, respectivamente.

Figura 81 - Classificação das Imagens de SR – Segunda mudança de Substituição



Fonte: autoria própria.

Figura 82 - Classificação das Imagens de SR – Terceira mudança de Substituição



Fonte: autoria própria.

Desse modo, examinando os resultados, começando pela análise dos gráficos de treinamento e validação, matrizes de confusão, relatório de classificação e métricas de classificação, é possível observar que o teste que mudou somente o primeiro “bloco” (primeira mudança) obteve os melhores resultados, com valores entre 93/94% de acerto, mantendo uma melhor “estabilidade” nos valores. Já a segunda e a terceira mudanças ficaram com valores muito próximos, entre 92 e 93%, tendo um desempenho um pouco pior em comparação com a primeira mudança.

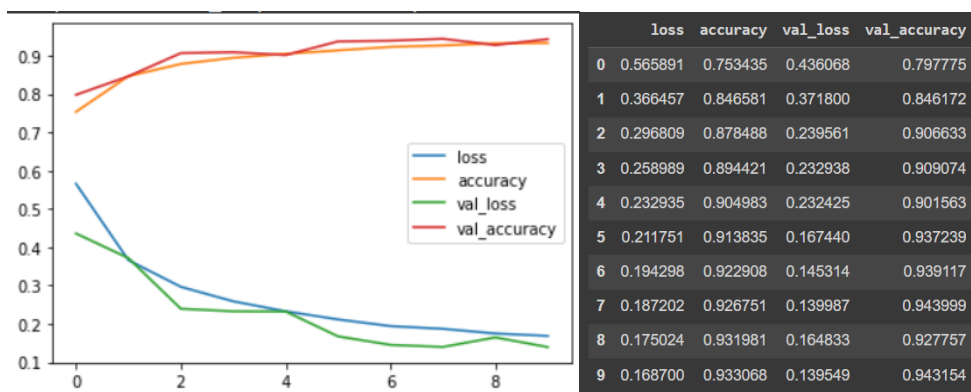
Comparando a classificação das imagens de LR e de SR, a primeira mudança (primeiro teste) acertou todas as imagens de LR, menos a classificação do linfócito (que foi classificado como neutrófilo), a segunda mudança classificou o monócito e o linfócito em neutrófilo, tendo somente dois acertos das quatro imagens e a terceira mudança também teve dois erros. Entretanto, o neutrófilo e o linfócito foram classificados como monócitos, sendo novamente o primeiro teste que obteve mais acertos comparado com as outras duas mudanças.

Já nas imagens de SR, a primeira mudança e a segunda mudança alcançaram dois acertos (cada), errando a classificação do linfócito e do monócito (classificados em eosinófilo e neutrófilo, respectivamente). Já a terceira mudança conseguiu três acertos, somente com o linfócito foi classificado como monócito. Assim, nas imagens de SR, a terceira mudança teve melhores resultados comparados com as duas primeiras, acertando mais classificações.

Desta forma, comparando cada resultado, percebe-se que a primeira mudança apresentou os melhores resultados na maioria dos testes. Por isso, a primeira mudança seguiu para os últimos dois testes, que faz também a troca de parte das imagens do *dataset* de validação.

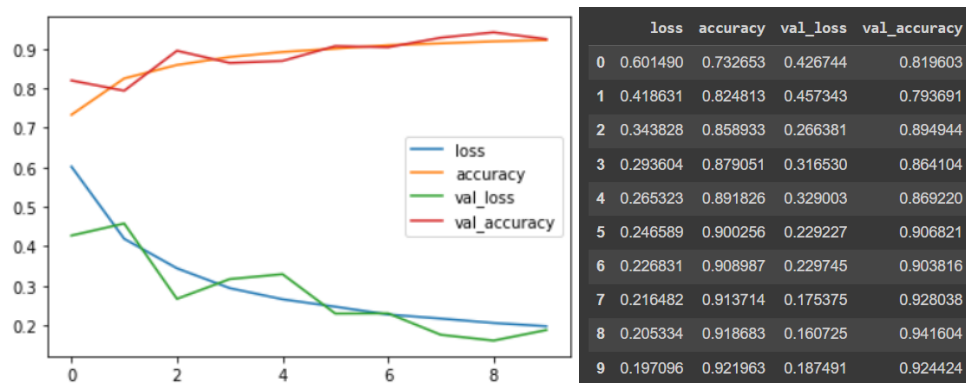
Assim, foram feitos o quarto e quinto testes de substituição, iniciando com os resultados para comparação. Na Figura 83 e 84 têm-se novamente os gráficos da acurácia e também de perda, tanto no treinamento quanto na validação, com os valores de cada época de treinamento. Já na Figura 85, observam-se as matrizes de confusão com o resultado de cada classe, observando de cima para baixo: quarto e quinto testes.

Figura 83 - Gráficos de Aprendizado e Valores por Época – Quarta mudança de Substituição



Fonte: autoria própria.

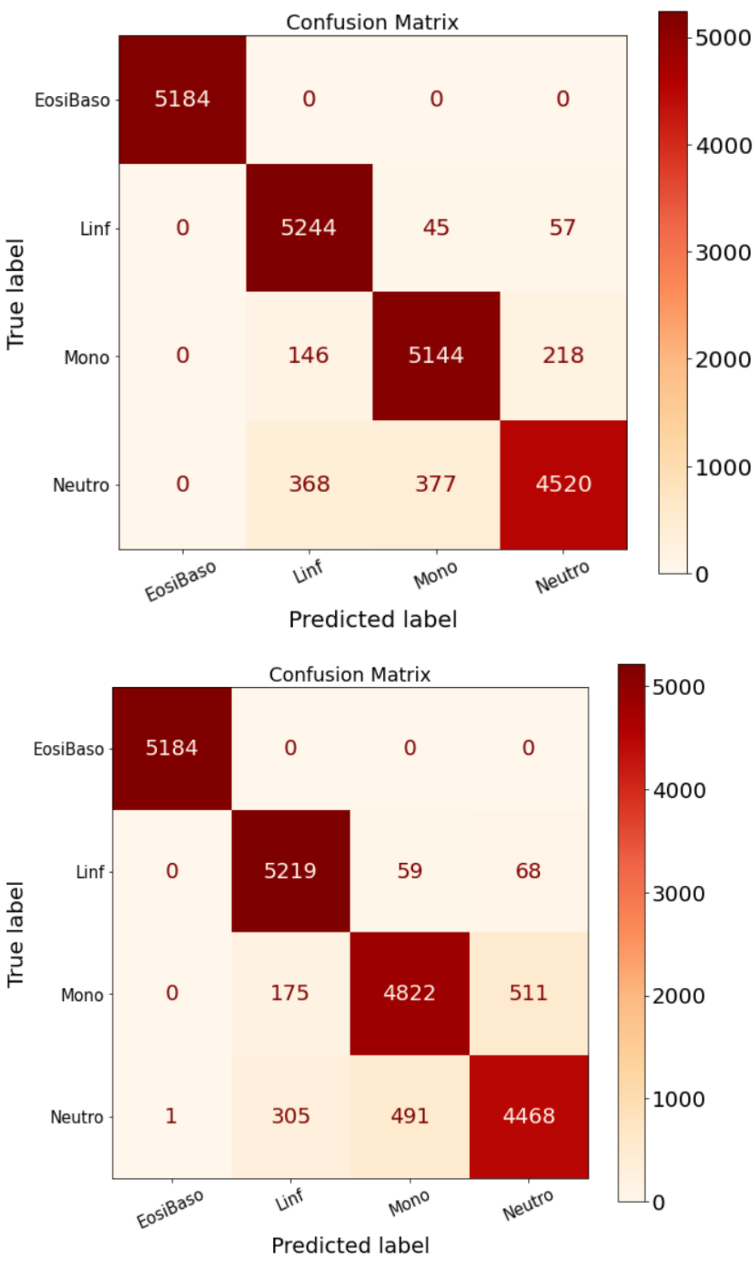
Figura 84 - Gráficos de Aprendizado e Valores por Época – Quinta mudança de Substituição



Fonte: autoria própria.

Como também foi analisado o Relatório de Classificação (Figuras 86 e 87), já explicado anteriormente, como mostrado na parte esquerda das figuras. Já na parte direita, estão as métricas (explicadas anteriormente), entretanto fazendo os cálculos considerando todas as classes.

Figura 85 - Matriz de Confusão – Quarta e Quinta mudanças de Substituição



Fonte: autoria própria.

Figura 86 - Relatório de Classificação e Métricas de Classificação – Quarta mudança de Substituição

	precision	recall	f1-score	support	
0	1.00	1.00	1.00	5184	Accuracy: 0.943153546448857
1	0.91	0.98	0.94	5346	Precision: 0.9438210721105211
2	0.92	0.93	0.93	5508	Sensitivity_recall: 0.943153546448857
3	0.94	0.86	0.90	5265	Specificity: 0.943153546448857
accuracy			0.94	21303	F1_score: 0.9426681336410595
macro avg	0.94	0.94	0.94	21303	
weighted avg	0.94	0.94	0.94	21303	

Fonte: autoria própria.

Figura 87 - Relatório de Classificação e Métricas de Classificação – Quinta mudança de Substituição

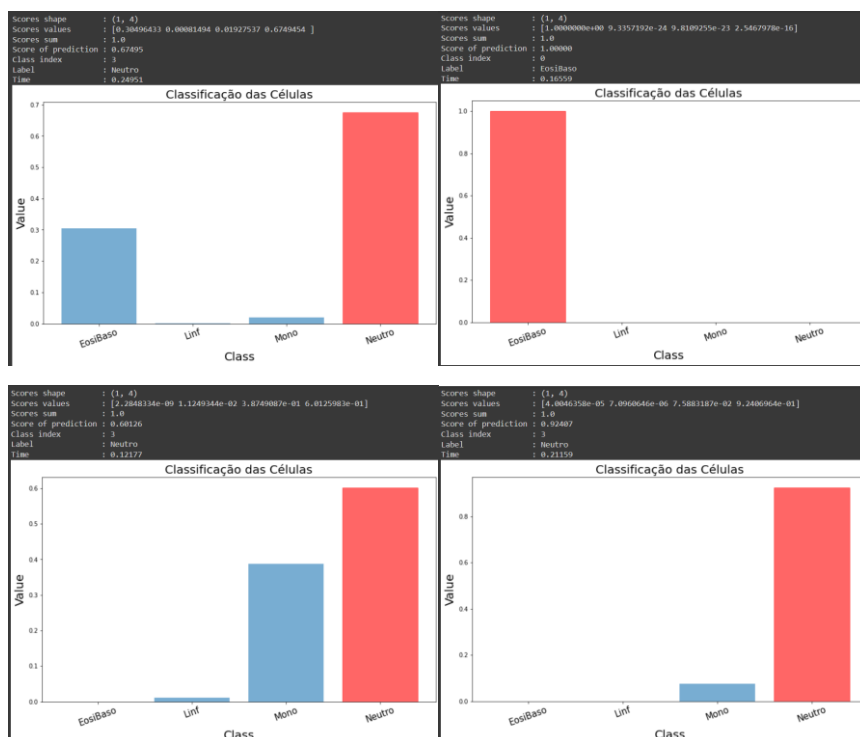
	precision	recall	f1-score	support	
0	1.00	1.00	1.00	5184	Accuracy: 0.9244237900765151
1	0.92	0.98	0.95	5346	Precision: 0.9239918491112032
2	0.90	0.88	0.89	5508	Sensitivity_recall: 0.9244237900765151
3	0.89	0.85	0.87	5265	Specificity: 0.9244237900765151
accuracy			0.92	21303	F1_score: 0.9238337487423612
macro avg	0.92	0.93	0.92	21303	
weighted avg	0.92	0.92	0.92	21303	

Fonte: autoria própria.

Após observar os valores das métricas, foram feitos os testes das imagens que são de LR e de SR, como demonstra as Figuras 88 e 89, já mostradas anteriormente.

A quarta mudança classificou as imagens de LR da seguinte maneira (observando a Figura 88 da esquerda para direita de cima para baixo): o linfócito foi classificado erroneamente como neutrófilo com aproximadamente 67%, como também o monócito foi classificado como neutrófilo com 60%. Já o eosinófilo foi classificado corretamente com 100% de certeza e igualmente o neutrófilo foi classificado corretamente com 92%.

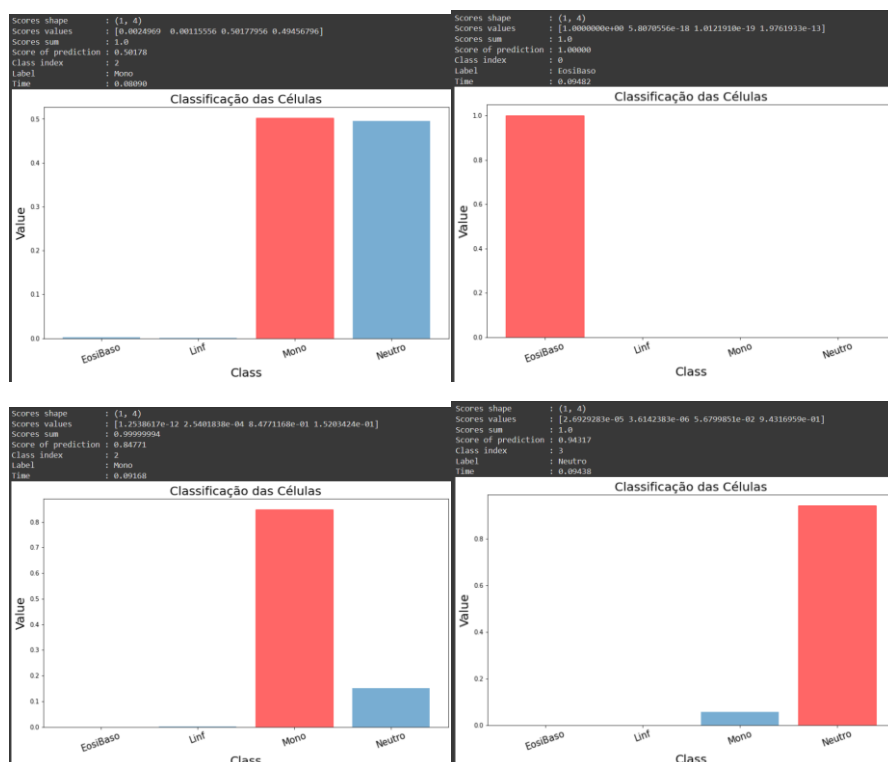
Figura 88 - Classificação das Imagens de LR – Quarta mudança de Substituição



Fonte: autoria própria.

Já a quinta mudança classificou as imagens de LR em (novamente observando a Figura 89 da esquerda para direita de cima para baixo): o linfócito foi classificado em monócito com aproximadamente 50%, o eosinófilo foi classificado corretamente com 100% de certeza, como também o neutrófilo foi classificado com 94% e também o monócito foi classificado aproximadamente 84%.

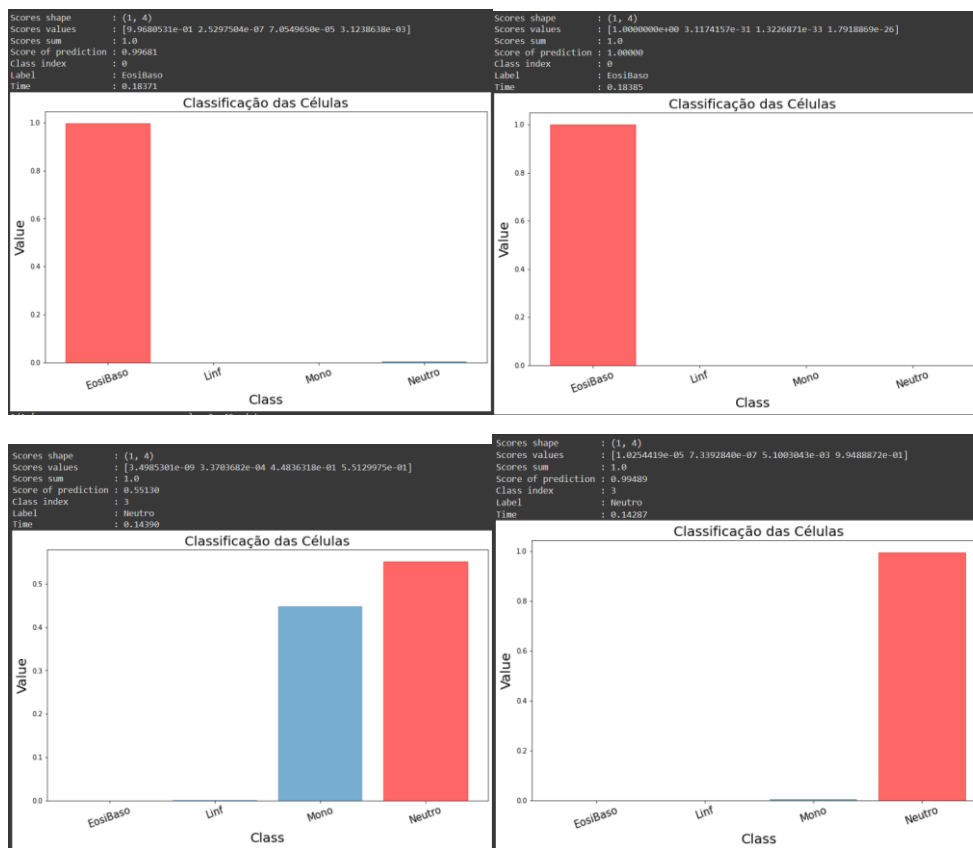
Figura 89 - Classificação das Imagens de LR – Quinta mudança de Substituição



Fonte: autoria própria.

Assim, com as imagens de SR, a quarta mudança classificou do seguinte modo (observando a Figura 90 da esquerda para direita de cima para baixo): o linfócito foi classificado como eosinófilo com 99%, como também o monócito foi classificado erroneamente como neutrófilo com 55%, o eosinófilo foi classificado corretamente com 100% de acerto e também o neutrófilo foi classificado com 99% de acerto.

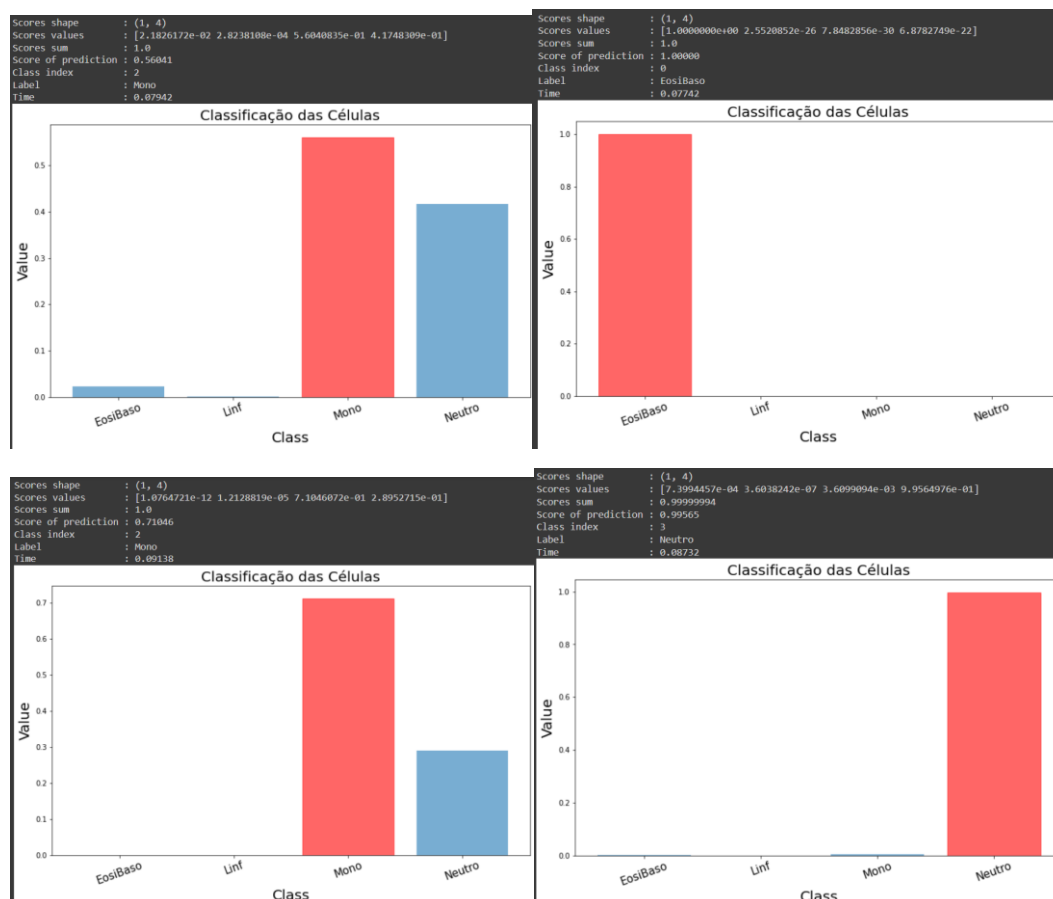
Figura 90 - Classificação das Imagens de SR – Quarta mudança de Substituição



Fonte: autoria própria.

Novamente, com as imagens de SR, entretanto agora com a quinta mudança classificou do seguinte modo (observando a Figura 91 da esquerda para direita de cima para baixo): o linfócito foi classificado como monócito com 56%, já as outras três células, eosinófilo, monócito e neutrófilo, foram classificados corretamente com 100%, 71% e 99%, respectivamente.

Figura 91 - Classificação das Imagens de SR – Quinta mudança de Substituição



Fonte: autoria própria.

Após observar todos os resultados, iniciando pela análise dos gráficos de treinamento e validação, relatório de classificação e métricas de classificação, chega-se à conclusão que a quarta mudança teve um resultado um pouco melhor comparado com a quinta mudança, com as métricas com valores de 94% e 92%, respectivamente. Comparando as matrizes de confusão, nota-se que em todas as classes, a quarta mudança teve melhores resultados, conseguindo acertar maior quantidade de imagens classificadas corretamente (somente a classe de eosinófilos que manteve o mesmo valor de 100% de acerto em ambas as mudanças).

Entretanto, analisando as imagens de LR e de SR, a quarta mudança obteve dois acertos (tanto nas imagens de LR e SR), errando a classificação do linfócito e do monócito, enquanto a quinta mudança conseguiu acertar três classificações, errando somente a classificação do linfócito. Desse modo, a quinta mudança atingiu melhores resultados no quesito de classificação das imagens de LR e de SR, no entanto em questão das métricas, o quarto teste obteve melhores resultados.

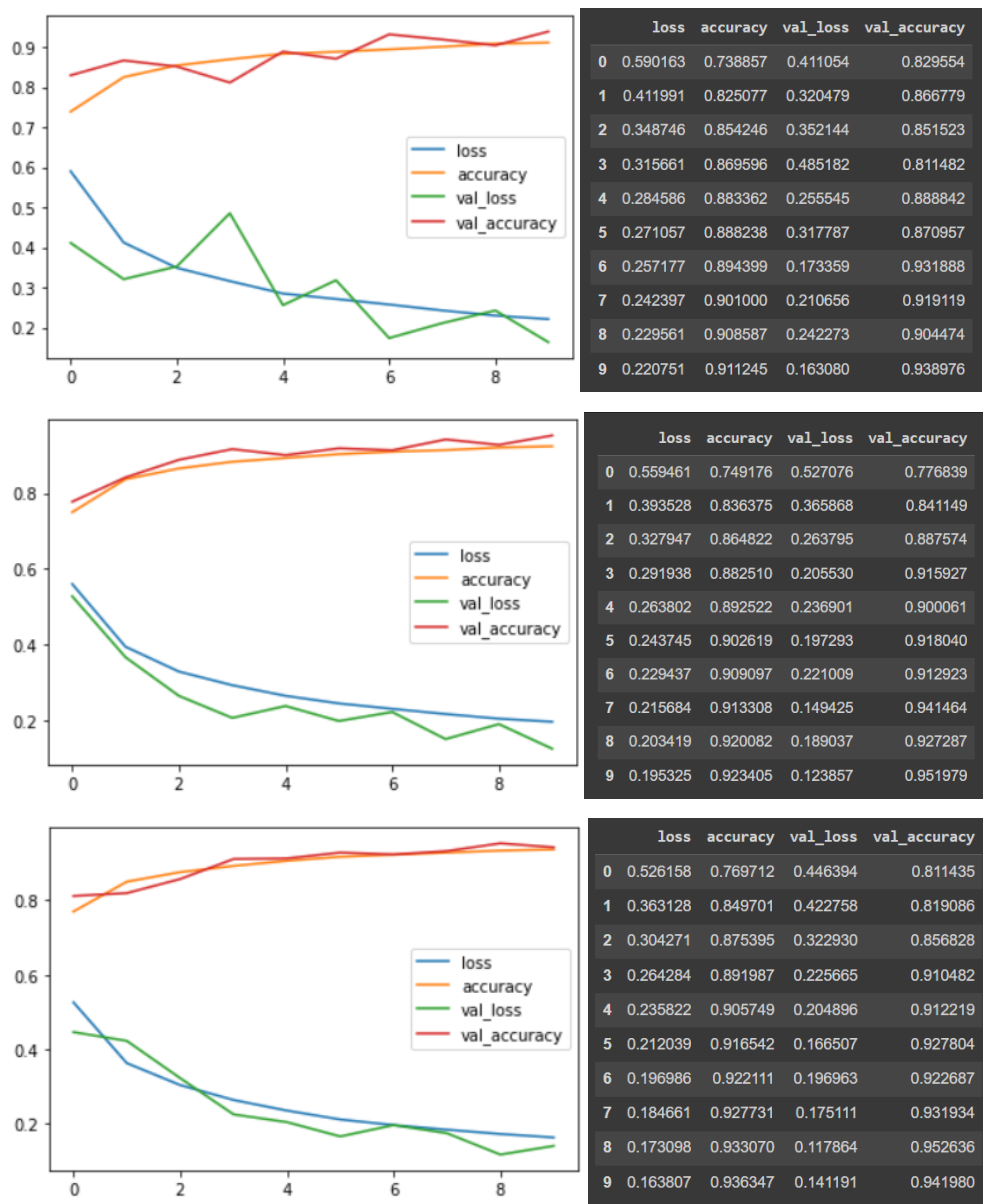
Confrontando todas as mudanças feitas (desde a primeira mudança até a quinta), a quarta mudança obteve os melhores resultados nas métricas de classificação. Porém, ainda obteve um resultado pior em relação aos algoritmos (tanto CNN quanto SVM) que usaram o *dataset* de imagens “originais” (sem ser de SR). Por outro lado, alcançou melhores resultados comparando com os algoritmos (novamente tanto CNN quanto SVM) que foram treinados com *dataset* 100% de imagens de SR. Ainda também sendo possível identificar que a CNN teve uma “oscilação” menor em comparação com a SVM sobre os valores das métricas de classificação dependendo do *dataset*, pois a CNN teve, com as imagens “originais”, um resultado próximo de 96% e a SVM próximo de 98%, já com o *dataset* 100% de SR, a CNN obteve aproximadamente 91%, já a SVM caiu para 89%.

Já comparando as classificações das imagens de LR e de SR, a maioria dos testes conseguiu atingir melhores resultados quando as imagens estavam com SR, sendo capaz de classificar de forma melhor as imagens (no caso, em realmente acertar a classificação ou até acertar com mais precisão/porcentagem) em comparação com as mesmas imagens, porém em LR. Assim, é possível afirmar que as imagens com SR ajudam, claramente, o classificador a classificar de forma correta e com mais precisão.

Assim, prosseguindo para os próximos testes de adição, observando a Figura 92, estão novamente os gráficos de acurácia e de perda (ambos de treinamento e de validação) juntamente com os valores de cada época (que já foram explicados anteriormente). Seguindo a ordem de cima para baixo está a primeira mudança até a terceira mudança.

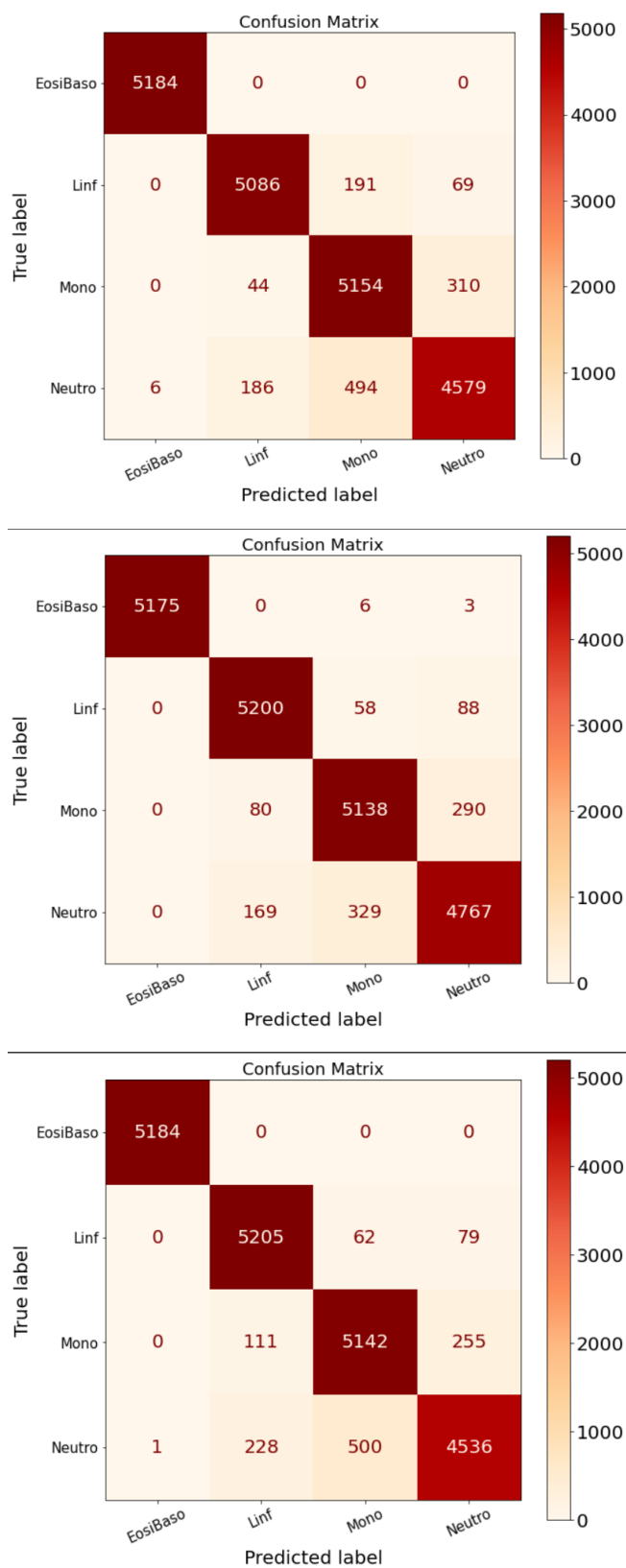
Na Figura 93 encontram-se as matrizes de confusão que, analisando de cima para baixo, têm-se primeira, segunda e terceira mudanças.

Figura 92 - Gráficos de Aprendizado e Valores por Época – Primeira, Segunda e Terceira mudanças de Adição



Fonte: autoria própria.

Figura 93 - Matriz de Confusão – Primeira, Segunda e Terceira mudanças de Adição



Fonte: autoria própria.

Foi possível também analisar o Relatório de Classificação (localizado na parte esquerda da Figura 94) e as métricas de classificação (situado na parte direita da Figura 94), ambas explicadas anteriormente. Examinando de cima para baixo estão o primeiro, segundo e terceiro testes.

Figura 94 - Relatório de Classificação e Métricas de Classificação – Primeira, Segunda e Terceira mudanças de Adição

	precision	recall	f1-score	support	
0	1.00	1.00	1.00	5184	Accuracy: 0.938975731117683
1	0.96	0.95	0.95	5346	Precision: 0.9396363129401653
2	0.88	0.94	0.91	5508	Sensitivity_recall: 0.938975731117683
3	0.92	0.87	0.90	5265	Specificity: 0.938975731117683
accuracy			0.94	21303	F1_score: 0.9389041639825165
macro avg	0.94	0.94	0.94	21303	
weighted avg	0.94	0.94	0.94	21303	

	precision	recall	f1-score	support	
0	1.00	1.00	1.00	5184	Accuracy: 0.9519785945641459
1	0.95	0.97	0.96	5346	Precision: 0.9518697783235839
2	0.93	0.93	0.93	5508	Sensitivity_recall: 0.9519785945641459
3	0.93	0.91	0.92	5265	Specificity: 0.9519785945641459
accuracy			0.95	21303	F1_score: 0.9518723742671767
macro avg	0.95	0.95	0.95	21303	
weighted avg	0.95	0.95	0.95	21303	

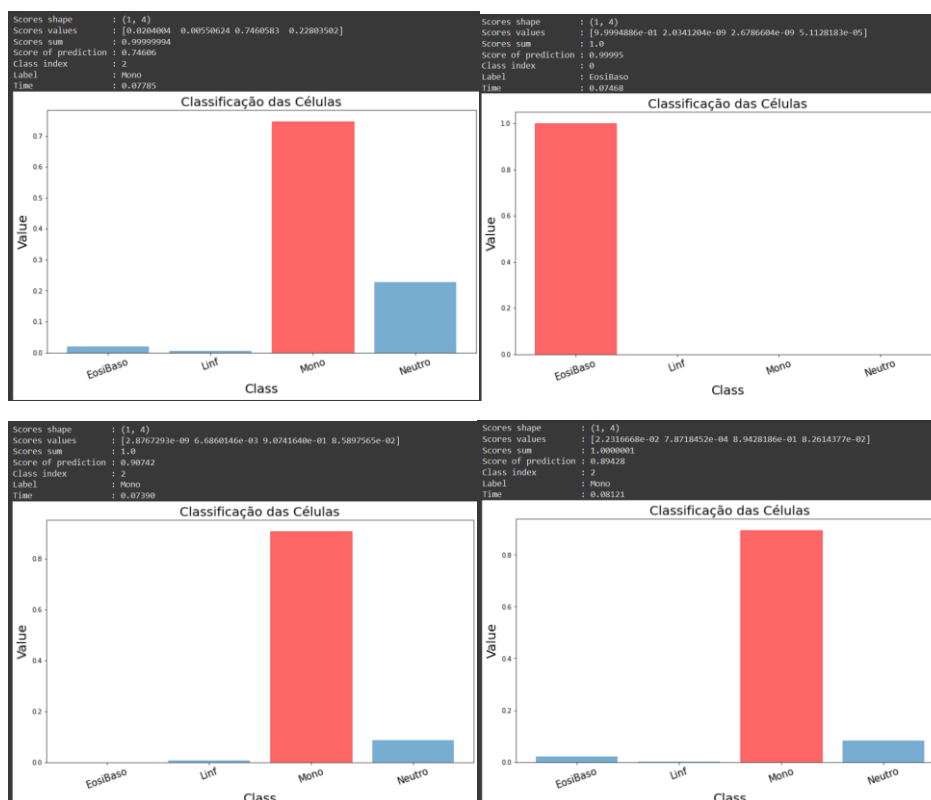
	precision	recall	f1-score	support	
0	1.00	1.00	1.00	5184	Accuracy: 0.9419800028165047
1	0.94	0.97	0.96	5346	Precision: 0.942183182983336
2	0.90	0.93	0.92	5508	Sensitivity_recall: 0.9419800028165047
3	0.93	0.86	0.90	5265	Specificity: 0.9419800028165047
accuracy			0.94	21303	F1_score: 0.9415932211911174
macro avg	0.94	0.94	0.94	21303	
weighted avg	0.94	0.94	0.94	21303	

Fonte: autoria própria.

Logo depois de analisar os resultados antecedentes, foram feitos os testes das imagens que são de LR e de SR, as imagens são as que se encontram nas Figuras 53 e 54, já mostradas anteriormente.

A primeira mudança catalogou as imagens de LR (Figura 95 da esquerda para direita, de cima para baixo) como: o linfócito foi classificado em monócito com aproximadamente 74% e o neutrófilo como monócito com 89%. Já o eosinófilo e o monócito foram classificados corretamente com 99% e 90%, respectivamente.

Figura 95 - Classificação das Imagens de LR – Primeira mudança de Adição

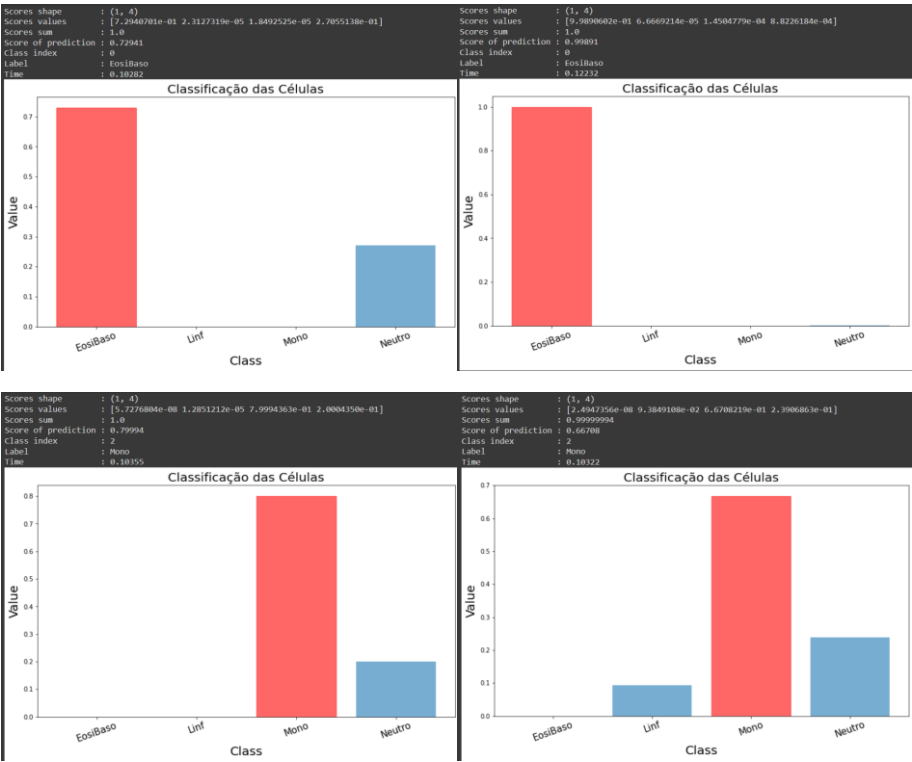


Fonte: autoria própria.

A segunda mudança teve os mesmos resultados de classificação que a primeira mudança, com dois acertos (eosinófilo com 99% e monócito com 79%) e dois erros (linfócito como eosinófilo com 72% e neutrófilo como monócito com 66%) nas mesmas imagens de LR (novamente observando a Figura 96 da esquerda para direita de cima para baixo).

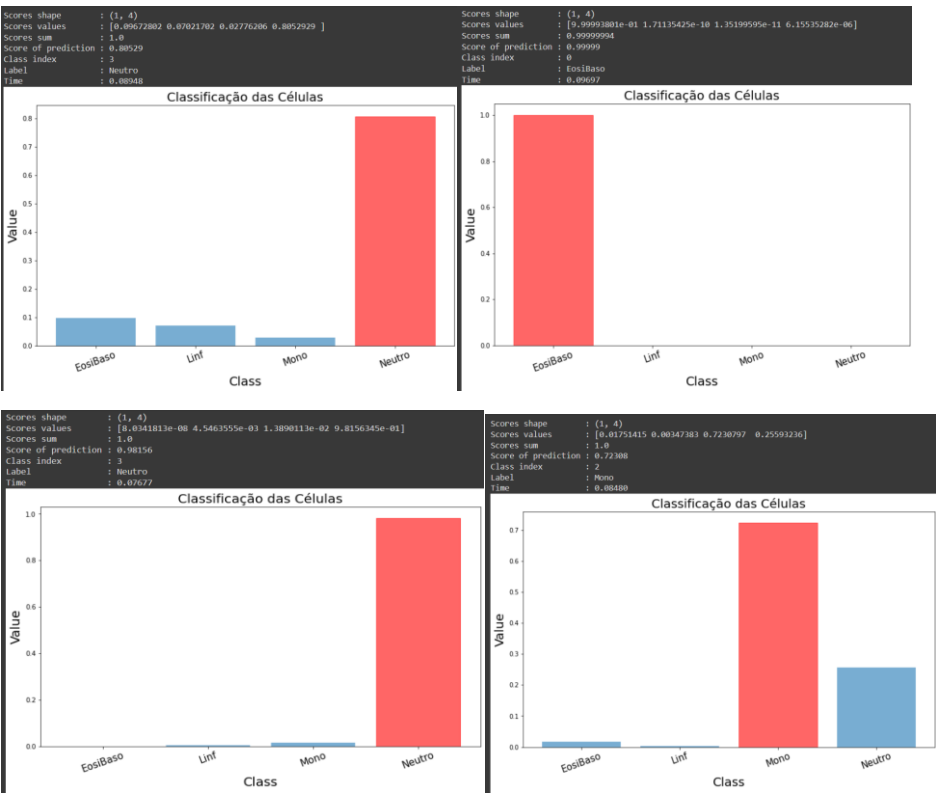
A terceira mudança obteve um pior desempenho, pois acertou somente uma (eosinófilo com 99%) das imagens e errou três (linfócito como neutrófilo com 80%, monócito como neutrófilo com 98% e neutrófilo como monócito com 72%), sendo possível observar na Figura 97 (da esquerda para direita e de cima para baixo).

Figura 96 - Classificação das Imagens de LR – Segunda mudança de Adição



Fonte: autoria própria.

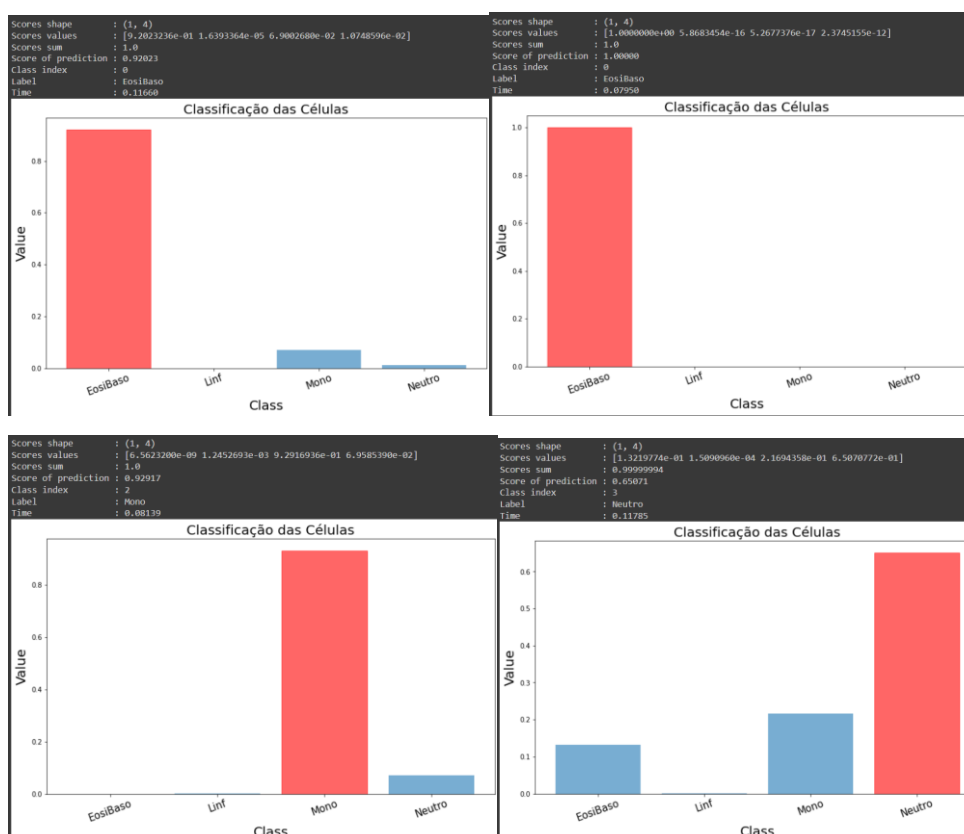
Figura 97 - Classificação das Imagens de LR – Terceira mudança de Adição



Fonte: autoria própria.

Para as imagens de SR, a primeira mudança teve três acertos, sendo o neutrófilo, o monócito e o eosinófilo, com 65%, 92% e 100%, respectivamente, e uma imprecisão, a qual foi o linfócito classificado como eosinófilo com 92%, como mostrado na Figura 98 (para ser observada da esquerda para direita e de cima para baixo).

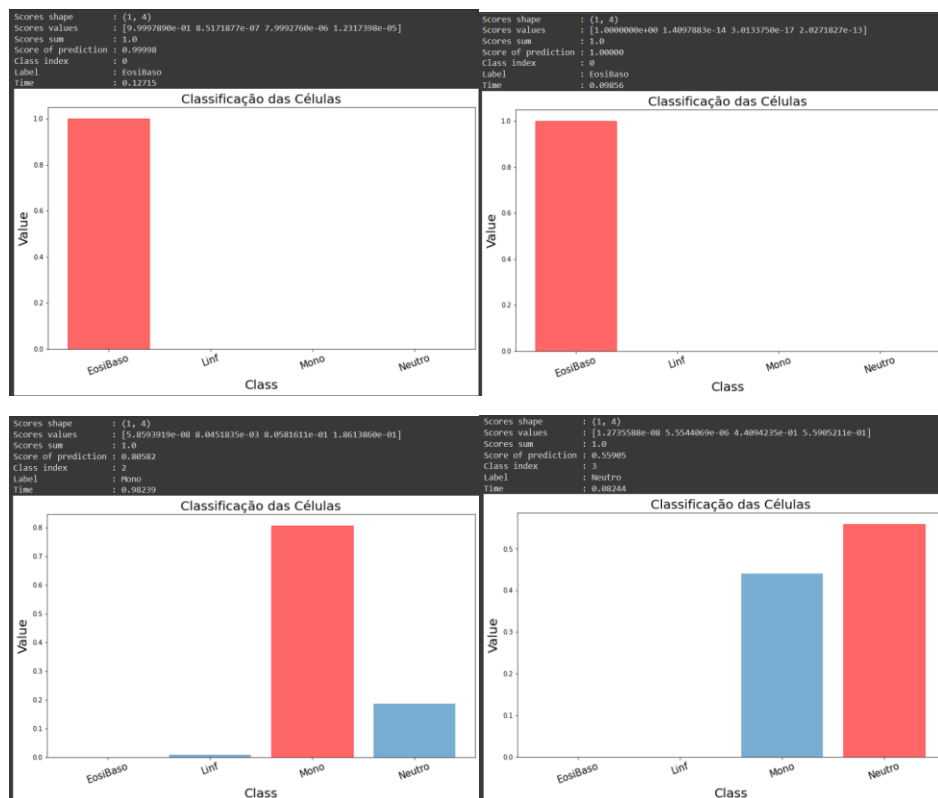
Figura 98 - Classificação das Imagens de SR – Primeira mudança de Adição



Fonte: autoria própria.

Agora, com a segunda mudança, obtive os mesmos resultados de classificação como a primeira mudança, com três acertos (eosinófilo com 100%, monócito com 80% e neutrófilo com 55%) e um erro (linfócito como eosinófilo com 99%) nas mesmas imagens de SR (novamente observando a Figura 99 da esquerda para direita de cima para baixo).

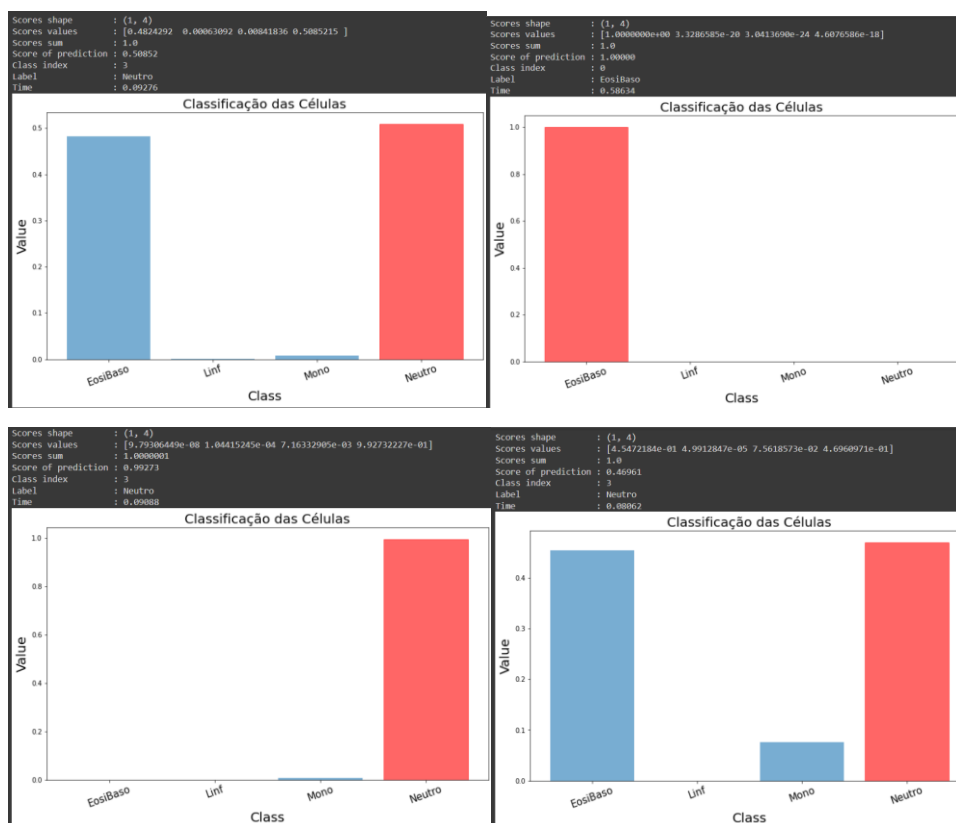
Figura 99 - Classificação das Imagens de SR – Segunda mudança de Adição



Fonte: autoria própria.

A terceira mudança com as imagens de SR teve um resultado pior comparado com a primeira e segunda mudanças, com dois acertos (eosinófilo com 100% e neutrófilo com 46%) e dois erros (linfócito como neutrófilo com 50% e monócito como neutrófilo com 99%), observando a Figura 100 da esquerda para direita e de cima para baixo.

Figura 100 - Classificação das Imagens de SR – Terceira mudança de Adição



Fonte: autoria própria.

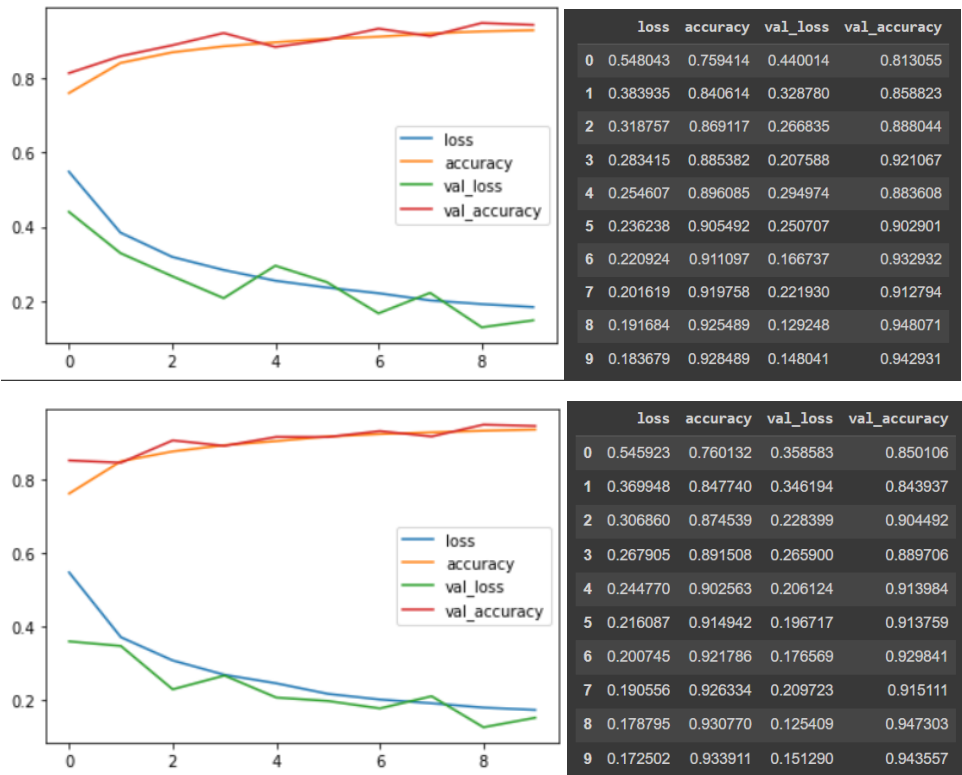
Dessa forma, analisando todos os resultados, partindo pela averiguação dos gráficos de treinamento e validação, matrizes de confusão, relatório de classificação e métricas de classificação, constata-se que a segunda mudança atingiu o melhor resultado com valores das métricas próximos de 95%, logo após, a terceira mudança com 94% e depois, a primeira com 93%.

Depois, examinando as imagens de LR, a primeira e a segunda mudanças conseguiram dois acertos, já a terceira acertou somente uma classificação. Seguindo para as imagens de SR, a primeira e a segunda mudanças alcançaram três acertos (um a mais que obtido nas imagens de LR) e a terceira acertou duas imagens. Assim, a segunda mudança atingiu melhores resultados (comparando as métricas de classificação e a classificação das imagens de LR e de SR). Por esse motivo, a segunda mudança seguiu para os últimos dois testes, que faz também a adição de parte das imagens do *dataset* de validação com imagens de SR.

Começando com as comparações dos resultados, na Figura 101 e na Figura 102, na primeira estão os gráficos de acurácia e de perda, em conjunto com os valores

de cada época, na segunda apresentam-se as matrizes de confusão, observando de cima para baixo as Figuras 101 e 102 está o quarto e logo após o quinto teste.

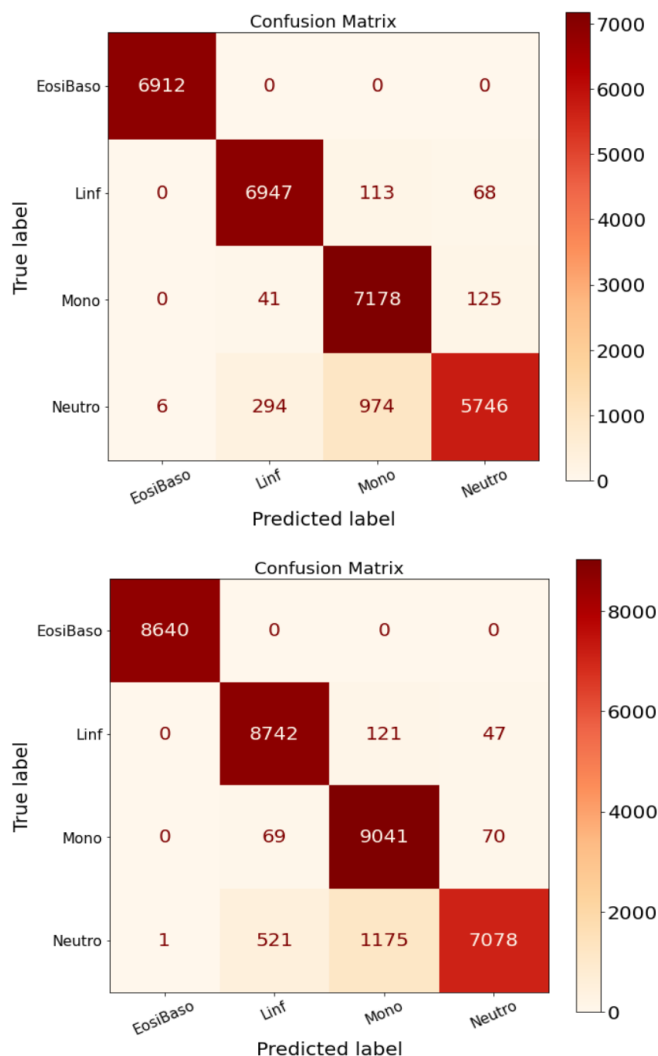
Figura 101 - Gráficos de Aprendizado e Valores por Época – Quarta e Quinta mudanças de Adição



Fonte: autoria própria.

Do mesmo modo, foi averiguado o Relatório de Classificação e as métricas de classificação, o primeiro posicionado no lado esquerdo da Figura 103 e o segundo situado na parte direita. Observando de cima para baixo na mesma Figura, têm-se o quarto e quinto testes.

Figura 102 - Matriz de Confusão – Quarta e Quinta mudanças de Adição



Fonte: autoria própria.

Figura 103 - Relatório de Classificação e Métricas de Classificação – Quarta e Quinta mudanças de Adição

	precision	recall	f1-score	support	
0	1.00	1.00	1.00	6912	Accuracy: 0.9429305731587101
1	0.95	0.97	0.96	7128	Precision: 0.9462078923711315
2	0.87	0.98	0.92	7344	Sensitivity_recall: 0.9429305731587101
3	0.97	0.82	0.89	7020	Specificity: 0.9429305731587101
accuracy			0.94	28404	F1_score: 0.9421749931170962
macro avg	0.95	0.94	0.94	28404	
weighted avg	0.95	0.94	0.94	28404	

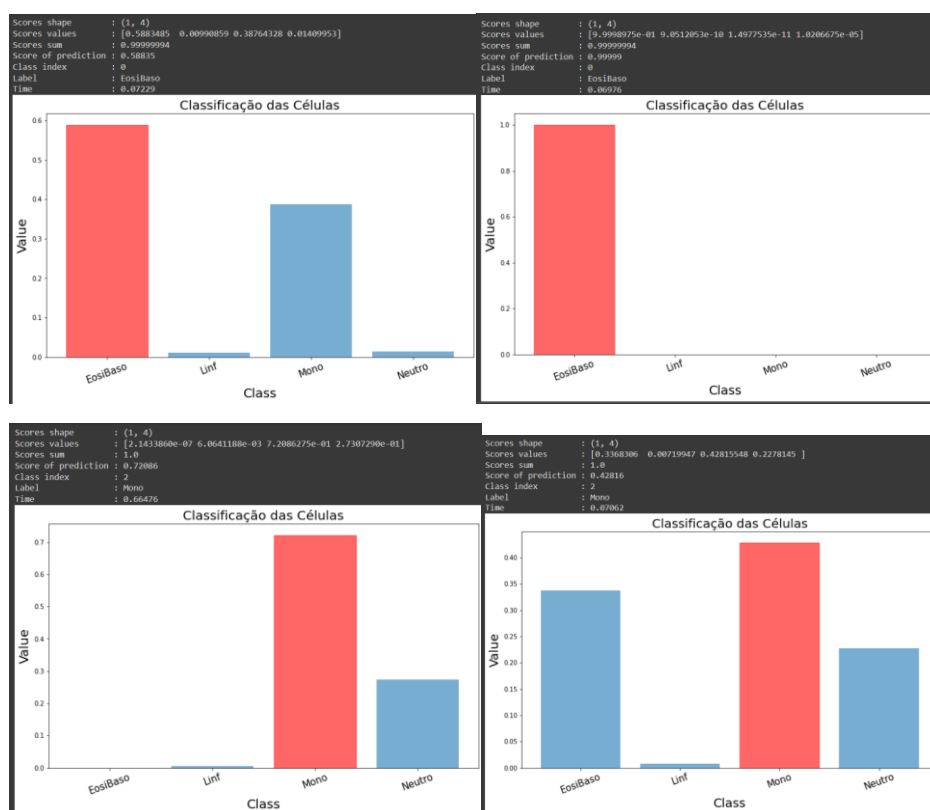
	precision	recall	f1-score	support	
0	1.00	1.00	1.00	8640	Accuracy: 0.9435572454583862
1	0.94	0.98	0.96	8910	Precision: 0.9476706445460931
2	0.87	0.98	0.93	9180	Sensitivity_recall: 0.9435572454583862
3	0.98	0.81	0.89	8775	Specificity: 0.9435572454583862
accuracy			0.94	35505	F1 score: 0.9424747973896846
macro avg	0.95	0.94	0.94	35505	
weighted avg	0.95	0.94	0.94	35505	

Fonte: autoria própria.

Logo após examinar os resultados das métricas, foram feitos os mesmos testes com as imagens de LR e de SR, já mostradas anteriormente.

A quarta mudança categorizou as imagens de LR (lendo a Figura 104 da esquerda para direita de cima para baixo) da seguinte forma: linfócito e neutrófilo classificados erroneamente como eosinófilo com 58% e monócito com 42%, respectivamente. Já o eosinófilo e o monócito foram categorizados corretamente com 99% e 72%, respectivamente.

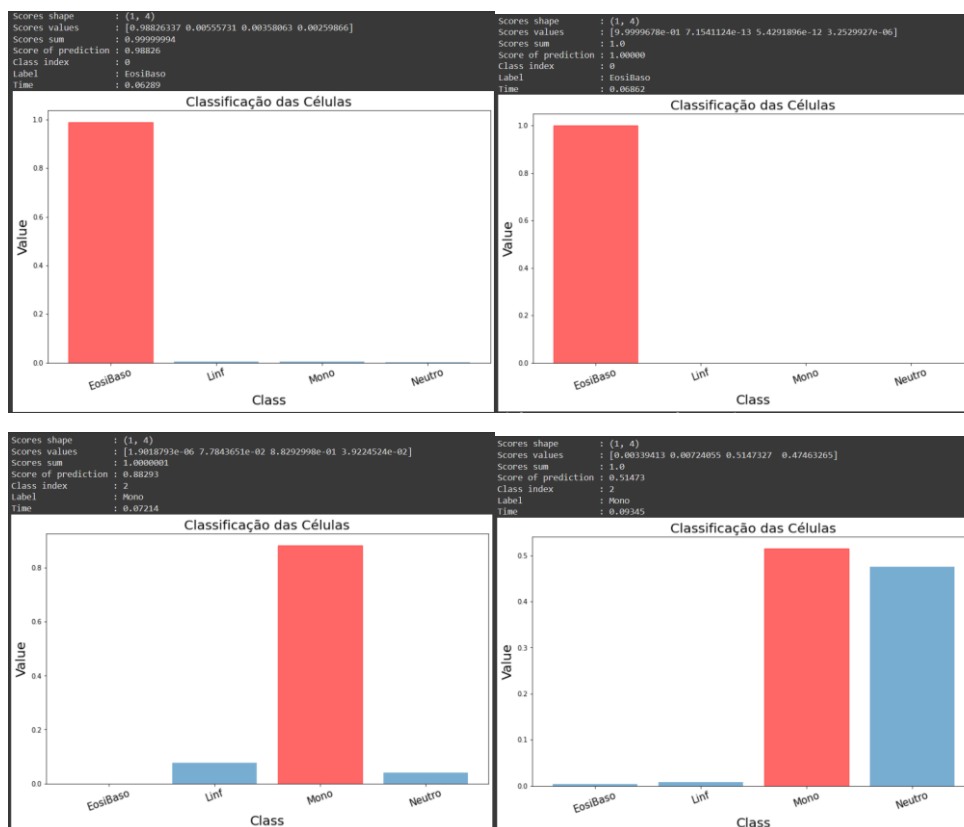
Figura 104 - Classificação das Imagens de LR – Quarta mudança de Adição



Fonte: autoria própria.

Igualmente, a quinta mudança atingiu as mesmas classificações nas imagens de LR que a quarta mudança, somente modificando as porcentagens (novamente observando a Figura 105 da esquerda para direita de cima para baixo). A imagem do linfócito que foi categorizada como eosinófilo com 98%, o neutrófilo foi classificado como monócito com 51%, eosinófilo com 100% e monócito com 88%.

Figura 105 - Classificação das Imagens de LR – Quinta mudança de Adição

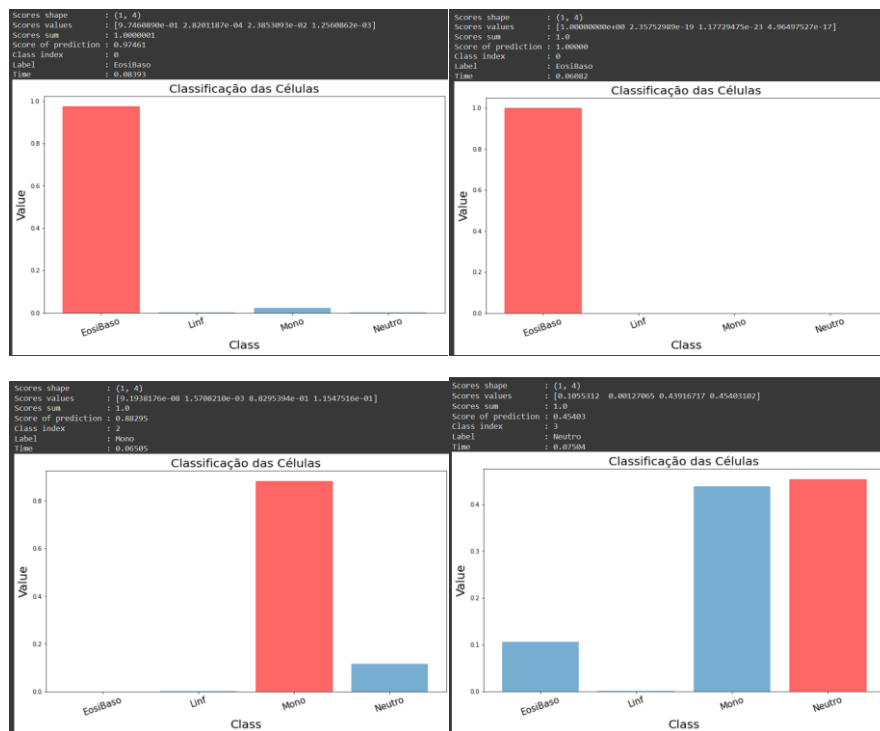


Fonte: autoria própria.

Deste modo, a quarta mudança, agora com as imagens de SR (observando a Figura 106 da esquerda para direita de cima para baixo): o linfócito foi classificado como eosinófilo com 97%, já o eosinófilo, o monócito e o neutrófilo foram classificados corretamente com 100%, 88% e 45%.

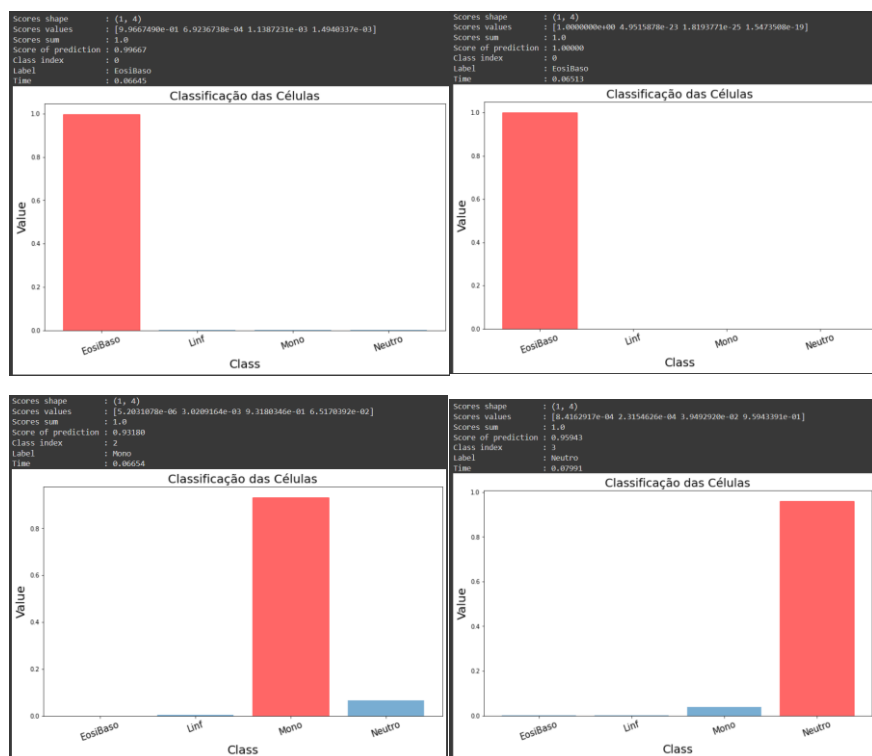
De novo, agora com as imagens de SR, a quinta mudança obteve novamente as mesmas classificações que a quarta mudança com as porcentagens: (novamente observando a Figura 107 da esquerda para direita de cima para baixo) a imagem do linfócito que foi categorizada como eosinófilo com 99%, já o neutrófilo, o monócito e o eosinófilo foram classificados de forma correta com 95%, 93% e 100%, respectivamente.

Figura 106 - Classificação das Imagens de SR – Quarta mudança de Adição



Fonte: autoria própria.

Figura 107 - Classificação das Imagens de SR – Quinta mudança de Adição



Fonte: autoria própria.

Assim, investigando os últimos resultados (quarta e quinta mudanças), começando pela análise dos gráficos de treinamento e validação, matrizes de confusão, relatório de classificação e métricas de classificação, conclui-se que a quinta mudança teve um resultado levemente melhor que a quarta mudança, sendo que a diferença entre elas foi de décimos (como é possível observar na Figura 103).

Assim como na análise das imagens de LR e de SR, tanto a quarta e a quinta mudanças tiveram as mesmas classificações (acertando duas na LR e três na SR). Entretanto, a quinta mudança obteve valores “melhores” nas porcentagens de classificação (acertando a classificação com mais porcentagem). Com isso, no quesito de classificação de imagens (sendo LR e SR), a quinta mudança se saiu razoavelmente melhor.

Contrapondo todas as mudanças realizadas na parte da adição (desde a primeira mudança até a quinta de adição), a segunda mudança teve os melhores resultados nas métricas de classificação, com valores próximos de 95%, sendo um resultado levemente pior comparado com os testes (tanto CNN quanto SVM, a primeira alcançou 96% e a segunda atingiu 98%) que usaram o *dataset* de imagens “originais” (que não são de SR). Em compensação, obteve melhores resultados comparando com os testes que utilizaram o *dataset* com 100% de imagens de SR (novamente tanto CNN quanto SVM). Assim como na substituição, na parte de classificação das imagens de LR e de SR, em todos os testes, alcançou melhores classificações (tanto em questão de acertos das classes das imagens quanto na parte de acertar com maior porcentagem) quando as imagens estavam com SR, em comparação com as mesmas imagens, entretanto em LR. Desse modo, novamente é comprovado quantitativamente que as imagens com SR auxiliam o classificador a classificar de forma certa e com mais precisão.

Finalmente, comparando todas as modificações feitas (tanto na parte de substituição quanto na parte de adição), a que obteve melhores valores na parte das métricas de classificação foi a segunda mudança do teste de adição, com valores de aproximadamente de 95%, ficando um pouco abaixo do valor de CNN com imagens 100% originais (que obteve 96%), como apresentado anteriormente. Já nas imagens de LR e de SR, na maioria dos testes, os classificadores conseguiram acertar mais as imagens de SR, acertar desde com melhor porcentagem até acertar de fato a classe

em que a imagem se encontra (em comparação com a LR), mostrando que a SR tem grande influência para ajudar na classificação das imagens.

Assim, o Quadro 12 faz um resumo dos resultados das métricas de classificação de alguns artigos abordados na seção da Pesquisa 01 e Pesquisa 02 em comparação com o próprio trabalho feito.

Quadro 12 - Comparação entre o trabalho próprio com outros artigos

Métricas/ Artigo	Acurácia	Precisão	Especificidade	Sensibilidade
Jiao <i>et al.</i> (2018)	99,3%	99%	Não disponível	98,4%
Grochowski <i>et al.</i> (2019)	94,56%	Não disponível	100%	89,13%
Wang <i>et al.</i> (2019)	90,85%	94,69%	Não disponível	Não disponível
Malkawi <i>et al.</i> (2020)	98,7%	98,8%	99,7%	98,6%
Mohamed <i>et al.</i> (2020)	92%	86%	95%	86%
Abdulhay <i>et al.</i> (2018)	Não disponível	95,3%	91,66%	100%
Trabalho Próprio	95,19%	95,18%	95,19%	95,19%

Fonte: autoria própria.

O Quadro 12 não apresenta os trabalhos de “destaque” tanto da Pesquisa 01 e da Pesquisa 02, pois da Pesquisa 01, que foi destacado o trabalho de Ma *et al.* (2018), entretanto somente são apresentadas as métricas para avaliar a SR. Já da Pesquisa 02, que realçou os trabalhos de Tom *et al.* (2019) e de Deshpande, Gite e Aluvalu (2021), porém o primeiro destaca somente métricas de SR e o segundo somente explica as técnicas dentro do trabalho.

Por fim, outro subtópico dentro de “Resultados” é o “Caso Necessário” para abordar sobre usar o Google Colab em Lotes ou Sem *dataset* específico. Entretanto, não foi necessário seu uso, uma vez que foi possível realizar os testes dentro dos parâmetros apresentados anteriormente.

5 DISCUSSÃO

Esta seção começa com as discussões acerca das buscas feitas, sendo que após a Pesquisa 01 foi percebido que havia uma “lacuna” no quesito algoritmos e códigos que abordassem a parte de melhoramento de resolução de imagens, por ser uma pesquisa inicial. Por isso, foi preciso ser promovida a elaboração de uma segunda pesquisa (com uma nova *query*), que além de abordar os códigos e algoritmos, aproveitou-se para pesquisar também sobre identificação e classificação das células sanguíneas que também faz parte do trabalho de pesquisa.

Assim, iniciou-se uma revisão, pois não se encontrou na literatura um estudo de revisão que sintetizasse em somente um local as diferentes técnicas de SR e de classificação de células sanguíneas. Com essa revisão, foram encontradas as melhores técnicas para serem aplicadas na SR e também para a classificação das células sanguíneas, com suas vantagens e desvantagens em somente um lugar.

Na próxima parte da discussão, foi abordado sobre as mudanças feitas no código de SR, focando, principalmente, na modificação do *Dropout*, que tem como objetivo de generalizar melhor a rede neural para que tenha menores chances de ocorrer o *Overfitting*. Houve resultados intrigantes em que com a adição dessa camada de *Dropout* as imagens que saíram foram extremamente afetadas, criando imagens muito distintas do que o esperado. Esses resultados podem ter ocorrido devido a uma generalização altíssima da rede a ponto de gerar imagens diferentes do que o aguardado.

Na sequência da discussão, foi passado para os resultados macro. Remetendo-se para o Quadro 12, percebe-se que a especificidade foi melhor no trabalho de Grochowski *et al.* (2019), com valor de 100%. Porém, a sensibilidade foi pior em comparação com todos os outros trabalhos registrados no Quadro 12. Já o trabalho de Jiao *et al.* (2018) apresentou os melhores resultados nas métricas de acurácia e de precisão, com 99,3% e 99%, respectivamente. Comparando-se, os trabalhos de Jiao *et al.* (2018) e de Malkawi *et al.* (2020) foram os que apresentaram melhores resultados nas métricas de classificação no geral, pois além de manterem as métricas em valores próximos entre 98% e 99%, não houve oscilações como no trabalho de Grochowski *et al.* (2019) e Mohamed *et al.* (2020). O primeiro teve a especificidade com 100%, porém a sensibilidade com 89,13%, e o segundo com a

especificidade com 95%. Entretanto, tiveram tanto a precisão e a sensibilidade com 86%, demonstrando essa variação entre as métricas.

O trabalho proposto permaneceu “balanceado” ou “equilibrado”, pois não apresentou nem os melhores resultados tão pouco os piores, estando em terceiro lugar nas métricas de acurácia e na especificidade, em quarto no quesito de precisão e de sensibilidade em comparação com os outros trabalhos apresentados no Quadro 12. Isso se deve, provavelmente, pois outros trabalhos apresentaram melhores ou maiores *datasets* para treinamento, ajudando a diversificar as imagens dos classificadores. Outra possibilidade pode ser devido ao tempo de treinamento ou o gasto computacional, podendo ser necessário mais tempo ou melhores máquinas para obter um resultado ainda mais alto.

Outro teste que foi feito, porém não foi possível comparar com outros trabalhos (pois os trabalhos não fizeram esse tipo de teste), foram os testes realizados com as imagens de LR e de SR, que demonstram, de fato, que usando a SR em imagens de LR, e inserindo no classificador, é possível distinguir muito melhor as imagens que estão com SR que as imagens de LR, provando que a SR consegue, sim, colaborar e facilitar a classificação da imagem de células sanguíneas.

6 CONCLUSÃO

A revisão se mostrou muito válida, pois além de preencher a lacuna que foi encontrada na literatura, possibilitou conhecer e comparar os resultados de diversas técnicas. Inclusive, foi possível selecionar as melhores técnicas e algoritmos para este trabalho, a qual a técnica selecionada foi a CNN, sendo que escolher o método, foi a principal conclusão da revisão.

Além disso, foi possível implementar a CNN conforme apresentado na literatura para aplicação para as imagens que foram utilizadas. Ademais, pode-se afirmar que os resultados dos artigos que vieram na revisão possibilitaram apresentar resultados que possibilitaram escolher os melhores algoritmos para aquilo se propõe esse trabalho.

Foi possível comparar os resultados com os de outros trabalhos, revelando que no geral a CNN proposta alcançou ótimos valores (acima de 95% nas métricas de classificação), exibindo diversos resultados enriquecedores para o trabalho de pesquisa desenvolvido.

Logo, este trabalho conseguiu investigar e comparar, com base nos resultados apresentados e métodos, o desempenho de algoritmos de classificação como um todo, sendo um trabalho de investigação sobre a aplicação da técnica de SR aplicadas a classificação de imagens de células de sangue. E confirmando que a SR, sim, permite melhorar a classificação de imagens de células sanguíneas em comparação com imagens de LR. Ademais, ajuda outros pesquisadores a encontrar os diferentes tipos de técnicas com suas vantagens e desvantagens nessa “super” revisão.

6.1 CONTRIBUIÇÃO CIENTÍFICA, TECNOLÓGICA E SOCIAL

Na contribuição científica, o método proposto apresenta um estudo sobre diferentes as técnicas que existem na literatura, podendo ocorrer a comparação entre os desempenhos.

Já na contribuição tecnológica, o algoritmo proposto é a própria contribuição.

E, por último, a contribuição social é ajudar os profissionais da saúde a entregar um diagnóstico mais preciso e rápido, com o intuito de oferecer um tratamento correto para salvar as vidas dos pacientes.

6.2 TRABALHOS FUTUROS

Com a conclusão desse trabalho, podem-se elencar os seguintes trabalhos futuros:

- Fazer testes com outros tipos de células, como as vermelhas e também as plaquetas.
- Treinar os classificadores com somente imagens de LR e comparar com os resultados com o dataset de 100% de SR, para observar qual a verdadeira melhora que ocorre nas métricas e nas classificações, com o auxílio do código de SR para criar os *datasets*.
- Usar outros tipos de algoritmos, inclusive para diferentes tipos de células
- Investigar generalização da rede *versus* métricas de classificação com a utilização de novas técnicas (além do *dropout*).
- Trabalhar com *dataset* maiores e com imagens diferentes, utilizando a metodologia *leave one subject out* para auxiliar na questão do viés.

REFERÊNCIAS

ABBAS, S. Microscopic images dataset for automation of RBCs counting. **Data in Brief**, v. 5, n., p. 35-40, 2015.

ABDULHAY, E.; MOHAMMED, M. A.; IBRAHIM, D. A.; ARUNKUMAR, N.; VENKATRAMAN, V. Computer Aided Solution for Automatic Segmenting and Measurements of Blood Leucocytes Using Static Microscope Images. **Journal of Medical Systems**, v. 42, n. 4, p. 12, 2018.

ABU-QASMIEH, I. Novel and Efficient Approach for Automated Separation, Segmentation, and Detection of Overlapped Elliptical Red Blood Cells. **Pattern Recognition and Image Analysis**, v. 28, n. 4, p. 792-804, 2018.

ACKERMANN, D.; SCHMITZ, G. Super-Resolution Velocity Estimation in Microvessels using Multiple Hypothesis Tracking. In: IEEE INTERNATIONAL ULTRASONICS SYMPOSIUM (IUS), 2015, Taipei. **Proceedings [...]**. Taipei: IEEE, 2015. p. 1-4.

AL-DULAIMI, K.; AL-SABAAWI, A.; RESEN, R. D.; STEPHAN, J. J.; ZWAYEN, A. Using Adapted JSEG Algorithm with Fuzzy C Mean for Segmentation and Counting of White Blood Cell and Nucleus Images. 2019 IEEE ASIA-PACIFIC CONFERENCE ON COMPUTER SCIENCE AND DATA ENGINEERING (CSDE), 2019, Melbourne. **Proceedings [...]**. Melbourne: IEEE, 2019. p. 1-7.

ALAM, M. M.; ISLAM, M. T. Machine learning approach of automatic identification and counting of blood cells. **Healthcare Technology Letters**, v. 6, n. 4, p. 103-108, 2019.

ALOMARI, Y. M.; ABDULLAH, S.; AZMA, R. Z.; OMAR, K. Automatic Detection and Quantification of WBCs and RBCs Using Iterative Structured Circle Detection Algorithm. **Computational and Mathematical Methods in Medicine**, v. 2014, p. 17, 2014.

ASHOUR, A. S.; WAHBA, M. A.; GHANNAM, R. A Cascaded Classification-Segmentation Reversible System for Computer-Aided Detection and Cells Counting in Microscopic Peripheral Blood Smear Basophils and Eosinophils Images. **IEEE Access**, v. 9, p. 78883-78901, 2021.

AYAS, S.; EKINCI, M. Microscopic image super resolution using deep convolutional neural networks. **Multimedia Tools and Applications**, v. 79, n. 21-22, p. 15397-15415, 2020.

BAGHLI, I.; NAKIB, A.; SELLAM, E.; BENAZZOUZ, M.; CHIKH, A.; PETIT, E. Hybrid framework based on Evidence theory for blood cell image segmentation. In: CONFERENCE ON MEDICAL IMAGING - BIOMEDICAL APPLICATIONS IN MOLECULAR, STRUCTURAL, AND FUNCTIONAL IMAGING, 2014, San Diego. **Proceedings [...]**. San Diego: SPIE, 2014. p. 314-321.

BAILO, O.; HAM, D.; SHIN, Y. M. Red blood cell image generation for data augmentation using Conditional Generative Adversarial Networks. In: 32ND IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2019, Long Beach. **Proceedings [...]**. Long Beach: IEEE, 2019. p. 1039-1048.

BANSAL, R.; RAJ, G.; CHOUDHURY, T. Blur Image Detection using Laplacian Operator and Open-CV. In: 5TH INTERNATIONAL CONFERENCE SYSTEM MODELING AND ADVANCEMENT IN RESEARCH TRENDS (SMART), 2016, Moradabad. **Proceedings [...]**. Moradabad: IEEE, 2016. p. 63-67.

BECATTI, M.; MANNUCCI, A.; ARGENTO, F. R.; GITTO, S.; VIZZUTTI, F.; MARRA, F.; TADDEI, N.; FIORILLO, C.; LAFFI, G. Super-Resolution Microscopy Reveals an Altered Fibrin Network in Cirrhosis: The Key Role of Oxidative Stress in Fibrinogen Structural Modifications. **Antioxidants**, v. 9, n. 8, p. 17, 2020.

BHANDARI, A.; STROHM, E. M.; KOLIOS, M. C.; RASKAR, R. Photoacoustic ToF Tomography of Blood Cells: From Mathematical Approximation to Super-resolution. In: 12TH INTERNATIONAL CONFERENCE ON SAMPLING THEORY AND APPLICATIONS (SAMPTA), 2017, Tallinn. **Proceedings [...]**. Tallinn: IEEE, 2017. p. 475-479.

CAI, X. H.; CHAN, R. H.; MORIGI, S.; SGALLARI, F. Framelet-Based Algorithm for Segmentation of Tubular Structures. In: 3RD INTERNATIONAL CONFERENCE ON SCALE SPACE AND VARIATIONAL METHODS IN COMPUTER VISION, 2011, Ein Gedi. **Proceedings [...]**. Ein Gedi: Springer-Verlag, 2011. p. 411.

CAI, X. H.; WALLIS, C. G. R.; CHAN, J. Y. H.; MCEWEN, J. D. Wavelet-based segmentation on the sphere. **Pattern Recognition**, v. 100, n., p. 15, 2020.

CLARO, M.; VOGADO, L.; VERAS, R.; SANTANA, A.; TAVARES, J.; SANTOS, J.; MACHADO, V. Convolution Neural Network Models for Acute Leukemia Diagnosis. In: 27TH INTERNATIONAL CONFERENCE ON SYSTEMS, SIGNALS AND IMAGE PROCESSING (IWSSIP), 2020, Niteroi. **Proceedings [...]**. Niteroi: IEEE, 2020. p. 63-68.

COOKE, C.; KIM, K.; XU, S. Q.; YAO, X.; YANG, X.; GLASS, C.; NEFF, J.; PITTMAN, P.; MCCALL, C.; JIANG, X. Y.; HORSTMAYER, R. Deep Optical Blood Analysis: COVID-19 Detection as a Case Study in Next Generation Blood Screening. **Modern Pathology**, v. 34, n. 2, p. 814-816, 2021.

DE CARVALHO, G. D. G.; SOKULSKI, C. C.; DA SILVA, W. V.; DE CARVALHO, H. G.; DE MOURA, R. V.; DE FRANCISCO, A. C.; VEIGA, C. P. Bibliometrics and systematic reviews: A comparison between the Proknow-C and the Methodi Ordinatio. **Journal of Informetrics**, v. 14, n. 3, p. 14, 2020.

DEEP LEARNING BOOK. **Como Funciona o Dropout?** 2022. Disponível em: <https://www.deeplearningbook.com.br/capitulo-23-como-funciona-o-dropout/>. Acesso em: 01 fev. 2022.

DENCKS, S.; PIEPENBROCK, M.; OPACIC, T.; KRAUSPE, B.; STICKELER, E.; KIESSLING, F.; SCHMITZ, G. Relative Blood Volume Estimation from Clinical Super-resolution US Imaging in Breast Cancer. In: IEEE INTERNATIONAL ULTRASONICS SYMPOSIUM (IUS), 2018, Kobe. **Proceedings [...]**. Kobe: IEEE, 2018. p. 1-4.

DESHPANDE, N. M.; GITE, S.; ALUVALU, R. A review of microscopic analysis of blood cells for disease detection with AI perspective. **Peerj Computer Science**, v. 7, p. 460, 2021.

DONG, C.; LOY, C. C.; HE, K. M.; TANG, X. O. Image Super-Resolution Using Deep Convolutional Networks. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 38, n. 2, p. 295-307, 2016.

PUCRS. **Atlas de Histologia em cores**. 2022. Disponível em: <https://editora.pucrs.br/edipucrs/acessolivro/livros/atlas-de-histologia/sangue.html>. Acesso em: 01 fev. 2022.

ELSALAMONY, H. A. Healthy and unhealthy red blood cell detection in human blood smears using neural networks. **Micron**, v. 83, p. 32-41, 2016.

ELSALAMONY, H. A. Anaemia cells detection based on shape signature using neural networks. **Measurement**, v. 104, p. 50-59, 2017.

EL-SHAFI, W.; MOHAMED, E. M.; ZEGHID, M.; ALI, A. M.; ALY, M. H. Hybrid Single Image Super-Resolution Algorithm for Medical Images. **Cmc-Computers Materials & Continua**, v. 72, n. 3, p. 4879-4896, 2022.

FALCON-RUIZ, A.; PAZ-VIERA, J.; SAHLI, H. Estimating Quality Bounds of JPEG 2000 Compressed Leukocytes Images. In: 2ND MEXICAN CONFERENCE ON PATTERN RECOGNITION, 2010, Puebla. **Proceedings [...]**. Puebla: Springer-Verlag Berlin, 2010. p. 107.

FAN, H. Y.; ZHANG, F. B.; XI, L.; LI, Z. Y.; LIU, G. H.; XU, Y. LeukocyteMask: An automated localization and segmentation method for leukocyte in blood smear images using deep neural networks. **Journal of Biophotonics**, v. 12, n. 7, p. 17, 2019.

FANG, Y.; YU, N.; WANG, R.; SU, D. An on-chip instrument for white blood cells classification based on a lens-less shadow imaging technique. **Plos One**, v. 12, n. 3, p., 2017.

FANG, Y.; YU, N. M.; LIU, Y. Classification of White Blood Cells by Convolution Neural Network in Lens-free Imaging System. In: 11TH INTERNATIONAL CONGRESS ON IMAGE AND SIGNAL PROCESSING, BIOMEDICAL ENGINEERING AND INFORMATICS (CISP-BMEI), 2018, Beijing. **Proceedings [...]**. Beijing: IEEE, 2018. p. 1-5.

FREJLICHOWSKI, D. Detection of Erythrocyte Cells in Microscopy Images. **Przegląd Elektrotechniczny**, v. 88, n. 10B, p. 264-267, 2012.

GASPARIN, A. T.; ARAUJO, C. I. F.; SCHMITT, P.; CARDOSO, M. R.; PERUSSOLO, M. C.; DE JESUS, T. C. S.; SANTIAGO, E. B.; SILVA, I. L. R.; DE SOUSA, R. G.; TENG, F. Z.; SEVERO, E. B.; RIBEIRO, V. H. A.; CARDOSO, M. A.; SILVA, F. D.; PERAZZOLI, C. R. D.; FARIAS, J. S. D.; DE ALMEIDA, B. M. M.; ROGAL, S. R.; FIGUEREDO, M. V. M. Hilab system, a new point-of-care hematology analyzer supported by the Internet of Things and Artificial Intelligence. **Scientific Reports**, v. 12, n. 1, p. 11, 2022.

GREENBAUM, A.; AKBARI, N.; FEIZI, A.; OZCAN, A. Field-Portable Lensfree Holographic Color Microscope for Telemedicine Applications. In: 3RD IEEE GLOBAL HUMANITARIAN TECHNOLOGY CONFERENCE (GHTC), 2013, San Jose. **Proceedings [...]**. San Jose: IEEE, 2013. p. 215.

GROCHOWSKI, M.; WASOWICZ, M.; MIKOLAJCZYK, A.; FICEK, M.; KULKA, M.; WROBEL, M. S.; JEDRZEJEWSKA-SZCZERSKA, M. Machine Learning System For Automated Blood Smear Analysis. **Metrology and Measurement Systems**, v. 26, n. 1, p. 81-93, 2019.

HABIBZADEH, M.; KRZYZAK, A.; FEVENS, T.; SADR, A. Counting of RBCs and WBCs in noisy normal blood smear microscopic images. In: CONFERENCE ON MEDICAL IMAGING 2011 - COMPUTER-AIDED DIAGNOSIS, 2011, Lake Buena Vista. **Proceedings [...]**. Lake Buena Vista: SPIE, 2011. p. 1009-1019.

HABIBZADEH, M.; KRZYZAK, A.; FEVENS, T. White Blood Cell Differential Counts Using Convolutional Neural Networks for Low Resolution Images. In: 12TH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND SOFT COMPUTING (ICAISC), 2013, Zakopane. **Proceedings [...]**. Zakopane: Springer-Verlag Berlin, 2013. p. 263-274.

HORÉ, A.; ZIOU, D. Image quality metrics: PSNR vs. SSIM. In: 2010 20TH INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION (ICPR 2010), 2010, Istanbul. **Proceedings [...]**. Istanbul: IEEE, 2010. p. 2366-2369.

HORTINELA, C. C.; FAUSTO, J. C.; VALIENTE, F. L.; DIVINA, P. D. C.; FELICES, J. P. T. Development of Abnormal Red Blood Cells Classifier Using Image Processing Techniques with Support Vector Machine. 10TH INTERNATIONAL CONFERENCE ON BIOMEDICAL ENGINEERING AND TECHNOLOGY (ICBET), 2020, Tokyo. **Proceedings [...]**. Tokyo: ACM, 2020. p. 17-21.

HUANG, X. W.; GUO, J. H.; WANG, X. L.; YAN, M.; KANG, Y. J.; YU, H. A Contact-Imaging Based Microfluidic Cytometer with Machine-Learning for Single-Frame Super-Resolution Processing. **Plos One**, v. 9, n. 8, p. 10, 2014.

HUANG, X. W.; JIANG, Y.; XU, H.; LIU, X.; HOU, H. W.; YAN, M.; YU, H. A Convolutional Neural Network based Single-frame Super-Resolution for Lensless Blood Cell Counting. In: 12TH IEEE BIOMEDICAL CIRCUITS AND SYSTEMS CONFERENCE (BIOCAS), 2016, Shanghai. **Proceedings [...]**. Shanghai: IEEE, 2016a. p. 168-171.

HUANG, X. W.; JIANG, Y.; LIU, X.; XU, H.; HAN, Z.; RONG, H. L.; YANG, H. P.; YAN, M.; YU, H. Machine Learning Based Single-Frame Super-Resolution Processing for Lensless Blood Cell Counting. **Sensors**, v. 16, n. 11, p. 16, 2016b.

JIANG, H.; LI, S.; LIU, W. H.; ZHENG, H. J.; LIU, J. H.; ZHANG, Y. Geometry-Aware Cell Detection with Deep Learning. **Msystems**, v. 5, n. 1, p. 11, 2020a.

JIANG, S.; GUAN, M. L.; WU, J. M.; FANG, G. C.; XU, X. Z.; JIN, D. Y.; LIU, Z.; SHI, K. B.; BAI, F.; WANG, S.; XI, P. Frequency-domain diagonal extension imaging. **Advanced Photonics**, v. 2, n. 3, p. 8, 2020b.

JIAO, Y.; SCHNEIDER, B. S.; REGENTOVA, E.; YANG, M.; ACM. Automated Quantification of White Blood Cells in Light Microscopy Muscle Images: Segmentation Augmented by CNN. In: 2ND INTERNATIONAL CONFERENCE ON VISION, IMAGE AND SIGNAL PROCESSING (ICVISIP), 2018, Las Vegas. **Proceedings [...]**. Las Vegas: ACM, 2018. p. 1-7.

KANAFIAH, S.; JUSMAN, Y.; ISA, N. A. M.; MOHAMED, Z. Radial-Based Cell Formation Algorithm for Separation of Overlapping Cells in Medical Microscopic Images. In: 1ST INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE AND COMPUTATIONAL INTELLIGENCE (ICCSICI), 2015, Jakarta. **Proceedings [...]**. Jakarta: Elsevier Science, 2015. p. 123-132.

KASAMSUMRAN, N.; CHANCHAITHONG, P.; WATTANAMONGKHOL, N.; PORA, W.; PUMRIN, S.; IEEE. Applying Faster R-CNN for Hematocytes Detection on Compound Microscope with Image Sensor Device and Multiple GPU Computation. In: 8TH INTERNATIONAL ELECTRICAL ENGINEERING CONGRESS (IEECON), 2020, Chiang Mai. **Proceedings [...]**. Chiang Mai: IEEE, 2020. p. 1-4.

KHASHMAN, A. Investigation of different neural models for blood cell type identification. **Neural Computing & Applications**, v. 21, n. 6, p. 1177-1183, 2012.

KIM, J.; KIM, J. Y.; JEON, S.; BAIK, J. W.; CHO, S. H.; KIM, C. Super-resolution localization photoacoustic microscopy using intrinsic red blood cells as contrast absorbers. **Light-Science & Applications**, v. 8, p. 11, 2019a.

KIM, J.; KIM, J. Y.; JEON, S.; BAIK, J. W.; CHO, S. H.; KIM, C. Agent-free High Speed Localization Photoacoustic Microscopy. In: CONFERENCE ON PHOTONS PLUS ULTRASOUND - IMAGING AND SENSING, 2019, San Francisco. **Proceedings [...]**. San Francisco: SPIE, 2019b. p. 82-87.

KOHLER, T.; BROST, A.; MOGALLE, K.; ZHANG, Q.; KOHLER, C.; MICHELSON, G.; HORNEGGER, J.; TORNOW, R. P. Multi-frame Super-resolution with Quality Self-assessment for Retinal Fundus Videos. In: 17TH INTERNATIONAL CONFERENCE ON MEDICAL IMAGE COMPUTING AND COMPUTER-ASSISTED INTERVENTION (MICCAI), 2014, Boston. **Proceedings [...]**. Boston: Springer, 2014, p. 650.

KUMAR, P. R.; SARKAR, A.; MOHANTY, S. N.; KUMAR, P. P. Segmentation of White Blood Cells using Image Segmentation Algorithms. In: 5TH INTERNATIONAL

CONFERENCE ON COMPUTING, COMMUNICATION AND SECURITY (ICCCS), 2020, Dayalpur. **Proceedings [...]**, Dayalpur: IEEE, 2020. p. 1-4.

LALITHA, V.; RAGHUL, G.; PREMKUMAR, A. R. Leukocyte Counting and Reporting Using Densenet-Deep Learning. ICPECTS 2020 - IEEE 2ND INTERNATIONAL CONFERENCE ON POWER, ENERGY, CONTROL AND TRANSMISSION SYSTEMS, 2020, Chennai. **Proceedings [...]**. Chennai: IEEE, 2020. p. 1-6.

LANE, W. J.; WESTHOFF, C. M.; GLEADALL, N. S.; AGUAD, M.; SMELAND-WAGMAN, R.; VEGE, S.; SIMMONS, D. P.; MAH, H. H.; LEBO, M. S.; WALTER, K.; SORANZO, N.; DI ANGELANTONIO, E.; DANESH, J.; ROBERTS, D. J.; WATKINS, N. A.; OUWEHAND, W. H.; BUTTERWORTH, A. S.; KAUFMAN, R. M.; REHM, H. L.; SILBERSTEIN, L. E.; GREEN, R. C.; BATES, D. W.; BLOUT, C.; CHRISTENSEN, K. D.; CIRINO, A. L.; HO, C. Y.; KRIER, J. B.; LEHMANN, L. S.; MACRAE, C. A.; MORTON, C. C.; PERRY, D. L.; SEIDMAN, C. E.; SUNYAEV, S. R.; VASSY, J. L.; SCHONMAN, E.; NGUYEN, T.; STEFFENS, E.; BETTING, W. N.; ARONSON, S. J.; CEYHAN-BIRSOY, O.; MACHINI, K.; MCLAUGHLIN, H. M.; AZZARITI, D. R.; TSAI, E. A.; BLUMENTHAL-BARBY, J.; FEUERMAN, L. Z.; MCGUIRE, A. L.; LEE, K.; ROBINSON, J. O.; SLASHINSKI, M. J.; DIAMOND, P. M.; DAVIS, K.; UBEL, P. A.; KRAFT, P.; ROBERTS, J. S.; GARBER, J. E.; HAMBUCH, T.; MURRAY, M. F.; KOHANE, I.; KONG, S. W.; MEDSEQ, P. Automated typing of red blood cell and platelet antigens: a whole-genome sequencing study. **The Lancet Haematology**, v. 5, n. 6, p. e241-e251, 2018.

LAVINA, B.; GAENGEL, K. New imaging methods and tools to study vascular biology. **Current Opinion in Hematology**, v. 22, n. 3, p. 258-266, 2015.

LEBEL, P.; DIAL, R.; VEMURI, V. N. P.; GARCIA, V.; DERISI, J.; GOMEZ-SJOBERG, R. Label-free imaging and classification of live *P. falciparum* enables high performance parasitemia quantification without fixation or staining. **Plos Computational Biology**, v. 17, n. 8, p. 29, 2021.

LI, J.; HUANG, Y. X.; ZHAO, H. Y.; TU, M.; CHEN, W. X. Superresolution measurement on the minute fluctuation of cell membrane. **Chinese Science Bulletin**, v. 51, n. 2, p. 143-147, 2006.

LI, J. W.; DAI, L.; YU, N. M.; LI, Z. P.; LI, S. J. Measurement of red blood cell size based on a lensless imaging system. **Biotechnology and Applied Biochemistry**, v. 68, n. 6, p. 1348-1356, 2021a.

LI, J. W.; DAI, L.; YU, N. M.; LI, Z. P.; LI, S. J. Elliptocyte detection technology based on super-resolution algorithms for a lensless imaging system. **Measurement Science and Technology**, v. 32, n. 2, p. 10, 2021b.

LIU, X.; HUANG, X. W.; JIANG, Y.; XU, H.; GUO, J.; HOU, H. W.; YAN, M.; YU, H. A Microfluidic Cytometer for Complete Blood Count With a 3.2-Megapixel, 1.1- μ m Pitch Super-Resolution Image Sensor in 65-nm BSI CMOS. **IEEE Transactions on Biomedical Circuits and Systems**, v. 11, n. 4, p. 794-803, 2017.

LIU, Y.; GEIPEL, M. M.; TIETZ, C.; BUETTNER, F. TIMELY: Improving labeling consistency in medical imaging for cell type classification. 24th European Conference on Artificial Intelligence, 2020, Santiago de Compostela. **Proceedings [...]**. Santiago de Compostela: IOS Press, 2020. p. 1858-1865.

LODDO, A.; PUTZU, L.; RUBERTO, C. D.; FENU, G. A Computer-Aided System for Differential Count from Peripheral Blood Cell Images. 12TH INTERNATIONAL CONFERENCE ON SIGNAL IMAGE TECHNOLOGY AND INTERNET-BASED SYSTEMS (SITIS), 2016, Naples. **Proceedings [...]**. Naples: IEEE. 2016. p. 112-118.

LONG, F.; PENG, J. J.; SONG, W. T.; XIA, X. B.; SANG, J. BloodCaps: A capsule network based model for the multiclassification of human peripheral blood cells. **Computer Methods and Programs in Biomedicine**, v. 202, p. 9, 2021.

LU, L.; ZHENG, Y.; CARNEIRO, G.; YANG, L. Deep learning and convolutional neural networks for medical image computing. **Advances in computer vision and pattern recognition**, v. 10, p. 978-973, 2017.

LUO, W.; ZHANG, Y.; FEIZI, A.; GOROCS, Z.; OZCAN, A. Pixel super-resolution using wavelength scanning. **Light-Science & Applications**, v. 5, p. 11, 2016.

MA, X.; YU, N. M.; FANG, Y.; WANG, D. F. Research on Single-Frame Super-Resolution Reconstruction Algorithm for Low Resolution Cell Images Based on Convolutional Neural Network. In: 3RD IEEE INTERNATIONAL CONFERENCE ON IMAGE, VISION AND COMPUTING (ICIVC), 2018, Chongqing. **Proceedings [...]**. Chongqing: IEEE, 2018. p. 369-373.

MAISELI, B.; MEI, J. Y.; GAO, H. J.; YIN, S. An Automatic and Cost-Effective Parasitemia Identification Framework for Low-End Microscopy Imaging Devices. In: INTERNATIONAL CONFERENCE ON MECHATRONICS AND CONTROL (ICMC), 2014, Jinzhou. **Proceedings [...]**. Jinzhou: IEEE, 2014. p. 2048-2053.

MALKAWI, A.; AL-ASSI, R.; SALAMEH, T.; SHEYAB, B.; ALQURAN, H.; ALQUDAH, A. M.; IEEE. White Blood Cells Classification Using Convolutional Neural Network Hybrid System. In: 5TH IEEE MIDDLE EAST AND AFRICA CONFERENCE ON BIOMEDICAL ENGINEERING (MECBME), 2020, Amman. **Proceedings [...]**. Amman: IEEE, 2020. p. 43-47.

MOALLEM, G.; POOSTCHI, M.; YU, H.; SILAMUT, K.; PALANIAPPAN, N.; ANTANI, S.; HOSSAIN, M. A.; MAUDE, R. J.; JAEGER, S.; THOMA, G. Detecting and Segmenting White Blood Cells in Microscopy Images of Thin Blood Smears. In: IEEE APPLIED IMAGERY PATTERN RECOGNITION WORKSHOP (AIPR), 2017, Washington. **Proceedings [...]**. Washington: IEEE, 2017. p. 1-8.

MOHAMED, E. H.; EL-BEHAIDY, W. H.; KHORIBA, G.; LI, J. Improved White Blood Cells Classification Based on Pre-trained Deep Learning Models. **Journal of Communications Software and Systems**, v. 16, n. 1, p. 37-45, 2020.

MOHAMMED, M. H.; DAWAY, H. G.; JOUDA, J. WBCs detection depending based on a binary conversion of the color component in a Ycbcr color space. In: 2020 IOP CONFERENCE SERIES - MATERIALS SCIENCE AND ENGINEERING, 2020, Thi-Qar. **Proceedings [...]**. Thi-Qar: IOP Publishing, 2020. p. 72081.

MORADIAMIN, M.; MEMARI, A.; SAMADZADEHAGHDAM, N.; KERMANI, S.; TALEBI, A. Computer aided detection and classification of acute lymphoblastic leukemia cell subtypes based on microscopic image analysis. **Microscopy Research and Technique**, v. 79, n. 10, p. 908-916, 2016.

MUDANYALI, O.; BISHARA, W.; OZCAN, A. Lensfree super-resolution holographic microscopy using wetting films on a chip. **Optics Express**, v. 19, n. 18, p. 17378-17389, 2011.

MUNDHRA, D.; CHELUVARAJU, B.; RAMPURE, J.; DASTIDAR, T. R. Analyzing Microscopic Images of Peripheral Blood Smear Using Deep Learning. In: 3RD MICCAI INTERNATIONAL WORKSHOP ON DEEP LEARNING IN MEDICAL IMAGE ANALYSIS (DLMIA) / 7TH INTERNATIONAL WORKSHOP ON MULTIMODAL LEARNING FOR CLINICAL DECISION SUPPORT (ML-CDS), 2017, Quebec. **Proceedings [...]**. Quebec: Springer International, 2017. p. 178-185.

NILSSON, B.; HEYDEN, A. Segmentation of complex cell clusters in microscopic images: Application to bone marrow samples. **Cytometry Part A**, v. 66A, n. 1, p. 24-31, 2005.

NILUFAR, S.; PERKINS, T. J. Learning a cost function for microscope image segmentation. 2014 36TH ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY (EMBC), 2014, Chicago. **Proceedings [...]**. Chicago: IEEE. p. 5506-5509.

OLIVEIRA, D. A. B.; PEREIRA, L. G. R.; BRESOLIN, T.; FERREIRA, R. E. P.; DOREA, J. R. R. A review of deep learning algorithms for computer vision systems in livestock. **Livestock Science**, v. 253, p. 15, 2021.

OPACIC, T.; DENCKS, S.; THEEK, B.; PIEPENBROCK, M.; ACKERMANN, D.; RIX, A.; LAMMERS, T.; STICKELER, E.; DELORME, S.; SCHMITZ, G.; KIESSLING, F. Motion model ultrasound localization microscopy for preclinical and clinical multiparametric tumor characterization. **Nature Communications**, v. 9, p. 13, 2018.

PENNANEN, P.; ALANNE, M. H.; FAZELI, E.; DEGUCHI, T.; NAREOJA, T.; PELTONEN, S.; PELTONEN, J. Diversity of actin architecture in human osteoclasts: network of curved and branched actin supporting cell shape and intercellular micrometer-level tubes. **Molecular and Cellular Biochemistry**, v. 432, n. 1-2, p. 131-139, 2017.

PETROVIC, N.; MOYA-ALCOVER, G.; JAUME-I-CAPO, A.; GONZALEZ-HIDALGO, M. Sick-cell disease diagnosis support selecting the most appropriate machine learning method: Towards a general and interpretable approach for cell morphology analysis from microscopy images. **Computers in Biology and Medicine**, v. 126, p. 14, 2020.

PIEPENBROCK, M.; BRIEDEN, A.; DENCKS, S.; SCHMITZ, G. Advancing the Feasible Microbubble Concentration in Super-Resolution. In: IEEE INTERNATIONAL ULTRASONICS SYMPOSIUM (IUS), 2019, Glasgow. **Proceedings [...]**. Glasgow: IEEE, 2019. p. 388-391.

PRINYAKUPT, J.; PLUEMPITIWIRIYAJEJ, C. Segmentation of white blood cells and comparison of cell morphology by linear and naïve Bayes classifiers. **Biomedical Engineering Online**, v. 14, n. 1, 2015.

QIN, F. W.; GAO, N. N.; PENG, Y.; WU, Z. Z.; SHEN, S. Y.; GRUDTSIN, A. Fine-grained leukocyte classification with deep residual learning for microscopic images. **Computer Methods and Programs in Biomedicine**, v. 162, p. 243-252, 2018.

QIU, W.; GUO, J. M.; LI, X.; XU, M. J.; ZHANG, M.; GUO, N.; LI, Q. Z. Multi-label Detection and Classification of Red Blood Cells in Microscopic Images. In: 8TH IEEE INTERNATIONAL CONFERENCE ON BIG DATA (BIG DATA), 2020, New York. **Proceedings [...]**. New York: IEEE, 2020, p. 4257-4263.

QUAN, Q.; WANG, J. X.; LIU, L. L. An Effective Convolutional Neural Network for Classifying Red Blood Cells in Malaria Diseases. **Interdisciplinary Sciences-Computational Life Sciences**, v. 12, n. 2, p. 217-225, 2020.

RAY, N. A concave cost formulation for parametric curve fitting: detection of leukocytes from intravital microscopy images. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, 2010, Hong Kong. **Proceedings [...]**. Hong Kong: IEEE, 2010. p. 53-56.

REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A.; You Only Look Once: Unified, Real-Time Object Detection. In: 2016 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2016, Seattle. **Proceedings [...]**. Seattle: IEEE, 2016. p. 779-788.

REENA, M. R.; AMEER, P. M. Localization and recognition of leukocytes in peripheral blood: A deep learning approach. **Computers in Biology and Medicine**, v. 126, p. 12, 2020.

SARRAFZADEH, O.; RABBANI, H.; DEHNAVI, A. M.; TALEBI, A. Circlet transform in cell and tissue microscopy. **Optics and Laser Technology**, v. 124, p. 16, 2020.

SAVKARE, S. S.; NAROTE, A. S.; NAROTE, S. P. Automatic blood cell segmentation using K-Mean clustering from microscopic thin blood images. THIRD INTERNATIONAL SYMPOSIUM ON COMPUTER VISION AND THE INTERNET, 2016, Jaipur. **Proceedings [...]**. Jaipur: ACM, 2016. p. 8-11.

SAYED, G. I.; SOLYMAN, M.; HASSANIEN, A. E. A novel chaotic optimal foraging algorithm for unconstrained and constrained problems and its application in white blood cell segmentation. **Neural Computing & Applications**, v. 31, n. 11, p. 7633-7664, 2019.

SELZER, R. H. Use of computers to improve biomedical image quality. In: FALL JOINT COMPUTER CONFERENCE, 1968, San Francisco. **Proceedings [...]**. San Francisco: ACM, 1968. p. 817-834.

SHAHIN, A. I.; GUO, Y. H.; AMIN, K. M.; SHARAWI, A. A. A novel white blood cells segmentation algorithm based on adaptive neutrosophic similarity score. **Health Information Science and Systems**, v. 6, p. 12, 2017.

SHAHZAD, M.; UMAR, A. I.; KHAN, M. A.; SHIRAZI, S. H.; KHAN, Z.; YOUSAF, W. Robust Method for Semantic Segmentation of Whole-Slide Blood Cell Microscopic Images. **Computational and Mathematical Methods in Medicine**, v. 2020, 2020.

SHAKARAMI, A.; MENHAJ, M. B.; MAHDAVI-HORMAT, A.; TARRAH, H. A fast and yet efficient YOLOv3 for blood cell detection. **Biomedical Signal Processing and Control**, v. 66, p. 11, 2021.

SHIRAZI, S. H.; UMAR, A. I.; HAQ, N. U.; NAZ, S.; RAZZAK, M. I. Accurate Microscopic Red Blood Cell Image Enhancement and Segmentation. In: 3RD INTERNATIONAL WORK-CONFERENCE ON BIOINFORMATICS AND BIOMEDICAL ENGINEERING (IWBBIO), 2015, Grenada. **Proceedings [...]**. Grenada: Springer International Publishing, 2015. p. 183-192.

SHIRAZI, S. H.; UMAR, A. I.; NAZ, S.; RAZZAK, M. I. Efficient leukocyte segmentation and recognition in peripheral blood image. **Technology and Health Care**, v. 24, n. 3, p. 335-347, 2016.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv**, n. 1409.1556, 2014.

SONG, H.; WANG, W. X. A new separation algorithm for overlapping blood cells using shape analysis. **International Journal of Pattern Recognition and Artificial Intelligence**, v. 23, n. 4, p. 847-864, 2009.

SONG, J.; XIAO, L.; LIAN, Z. Contour-Seed Pairs Learning-Based Framework for Simultaneously Detecting and Segmenting Various Overlapping Cells/Nuclei in Microscopy Images. **IEEE Transactions on Image Processing**, v. 27, n. 12, p. 5759-5774, 2018.

SUN, J. S.; CHEN, Q.; ZUO, C.; ZHANG, L. Resolution-improved Fourier ptychographic microscopy using high-numerical-aperture condenser. In: ANNUAL CONFERENCE OF THE CHINESE-SOCIETY-FOR-OPTICAL-ENGINEERING (CSOE) ON APPLIED OPTICS AND PHOTONICS CHINA (AOPC) - OPTICAL SENSING AND IMAGING TECHNOLOGY AND APPLICATIONS, 2017, Beijing. **Proceedings [...]**. Beijing: SPIE, 2017. p. 595-598.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCHE, V.; RABINOVICH, A. Going deeper with convolutions. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2015, Boston. **Proceedings [...]**. Boston: IEEE, 2015. p. 1-9.

TESSEMA, A. W.; MOHAMMED, M. A.; SIMEGN, G. L.; KWA, T. C. Quantitative analysis of blood cells from microscopic images using convolutional neural network. **Medical & Biological Engineering & Computing**, v. 59, n. 1, p. 143-152, 2021.

TOM, F.; SHARMA, H.; MUNDHRA, D.; DASTIDAR, T. R.; SHEET, D.; Learning a deep convolution network with turing test adversaries for microscopy image super resolution. In: 16TH IEEE INTERNATIONAL SYMPOSIUM ON BIOMEDICAL IMAGING (ISBI), 2019, Venice. **Proceedings [...]**. Venice: IEEE, 2019. p. 1391-1394.

TOMCZAK, A.; ILIC, S.; MARQUARDT, G.; ENGEL, T.; FORSTER, F.; NAVAB, N.; ALBARQOUNI, S. Multi-Task Multi-Domain Learning for Digital Staining and Classification of Leukocytes. **IEEE Transactions on Medical Imaging**, v. 40, n. 10, p. 2897-2910, 2021.

TOZETTI, P. B.; PINTO, F. E.; PEREIRA NETO, E.; ANDRADE, T. U.; ENDRINGER, D. C.; LENZ, D. A reproducible differential blood cell count without a cytometer or specific markers. **Journal of Morphological Sciences**, v. 32, n. 2, p. 78-81, 2015.

VIDYASARASWATHI, H. N.; HANUMANTHARAJU, M. C. Review of various histogram based medical image enhancement techniques. 2015 INTERNATIONAL CONFERENCE ON ADVANCED RESEARCH IN COMPUTER SCIENCE ENGINEERING & TECHNOLOGY (ICARCSET 2015), 2015, Unnao. **Proceedings [...]**. Unnao: ACM, 2015. p. 1-6.

WANG, Q. W.; BI, S. S.; SUN, M. L.; YANG, S. B.; CHEN, L. K.; WANG, J. Z. CNN-based Leukocyte Detection for Microscopy Imaging at 10x Magnification Objective Lens. In: 12TH INTERNATIONAL CONFERENCE ON MACHINE VISION (ICMV), 2019, Amsterdam. **Proceedings [...]**. Amsterdam: SPIE, 2019. p. 449-506.

WANG, S. T.; MIN, W. A new detection algorithm (NDA) based on fuzzy cellular neural networks for white blood cell detection. **IEEE Transactions on Information Technology in Biomedicine**, v. 10, n. 1, p. 5-10, 2006.

WANG, Z. H.; CHEN, J.; HOI, S. C. H. Deep Learning for Image Super-Resolution: A Survey. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 43, n. 10, p. 3365-3387, 2021.

WEI, X. D.; CAO, Y. P.; FU, G. K.; WANG, Y. P. A counting method for complex overlapping erythrocytes-based microscopic imaging. **Journal of Innovative Optical Health Sciences**, v. 8, n. 6, p. 11, 2015.

WEI, X. D.; CAO, Y. P. Automatic counting method for complex overlapping erythrocytes based on seed prediction in microscopic imaging. **Journal of Innovative Optical Health Sciences**, v. 9, n. 5, p. 9, 2016.

WESTMORELAND, D.; SHAW, M.; GRIMES, W.; METCALF, D. J.; BURDEN, J. J.; GOMEZ, K.; KNIGHT, A. E.; CUTLER, D. F. Super-resolution microscopy as a potential approach to diagnosis of platelet granule disorders. **Journal of Thrombosis and Haemostasis**, v. 14, n. 4, p. 839-849, 2016.

WIJESINGHE, C. B.; WICKRAMARACHCHI, D. N.; KALUPAHANA, I. N.; DE SERAM, L. R.; SILVA, I. D.; NANAYAKKARA, N. D. Fully Automated Detection and Classification of White Blood Cells. In: 42ND ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE-ENGINEERING-IN-MEDICINE-AND-BIOLOGY-SOCIETY (EMBC), 2020, Montreal. **Proceedings [...]**. Montreal: IEEE, 2020, p. 1816-1819.

XIA, T. C.; JIANG, R.; FU, Y. Q.; JIN, N. L.; IOP. Automated Blood Cell Detection and Counting via Deep Learning for Microfluidic Point-of-Care Medical Devices. In: 3RD INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE APPLICATIONS AND TECHNOLOGIES (AIAAT), 2019, Beijing. **Proceedings [...]**. Beijing: IOP Publishing, 2019. p. 12048.

XIA, T. C.; FU, Y. Q.; JIN, N. L.; CHAZOT, P.; ANGELOV, P.; JIANG, R. AI-enabled Microscopic Blood Analysis for Microfluidic COVID-19 Hematology. In: 5TH INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND APPLICATIONS (ICCIA), 2020, Beijing. **Proceedings [...]**. Beijing: IEEE, 2020. p. 98-102.

XU, M. J.; PAPAGEORGIOU, D. P.; ABIDI, S. Z.; DAO, M.; ZHAO, H.; KARNIADAKIS, G. E. A deep convolutional neural network for classification of red blood cells in sickle cell anemia. **Plos Computational Biology**, v. 13, n. 10, p. 27, 2017.

YANG, D. X.; ZHAO, H. D.; HAN, T. C.; KANG, Q.; MA, J. C.; LU, H. Y. Leukocyte subtypes identification using bilinear self-attention convolutional neural network. **Measurement**, v. 173, p. 10, 2021.

YAO, X. F.; SUN, K.; BU, X. X.; ZHAO, C. Y.; JIN, Y. Classification of white blood cells using weighted optimized deformable convolutional neural networks. **Artificial Cells Nanomedicine and Biotechnology**, v. 49, n. 1, p. 147-155, 2021.

YORK, A. G.; CHANDRIS, P.; NOGARE, D. D.; HEAD, J.; WAWRZUSIN, P.; FISCHER, R. S.; CHITNIS, A.; SHROFF, H. Instant super-resolution imaging in live cells and embryos via analog image processing. **Nature Methods**, v. 10, n. 11, p. 1122-1126, 2013.

YOUSEFI, S.; QIN, J.; WANG, R. K. Super-resolution spectral estimation of optical micro-angiography for quantifying blood flow within microcirculatory tissue beds in vivo. **Biomedical Optics Express**, v. 4, n. 7, p. 1214-1228, 2013.

ZALEVSKY, Z.; BORKOWSKI, A.; MAROM, E.; JAVIDI, B.; BEIDERMAN, Y.; MICO, V.; GARCIA, J. Recent advances in the field of super resolved imaging and sensing. In: CONFERENCE ON OPTICAL MEASUREMENT SYSTEMS FOR INDUSTRIAL INSPECTION VII, 2011, Munich. **Proceedings [...]**. Munich: SPIE, 2011. p. 143-152.

ZAMANI, F.; SAFABAKHSH, R. An unsupervised GVF snake approach for white blood cell segmentation based on nucleus. In: 8TH INTERNATIONAL

CONFERENCE ON SIGNAL PROCESSING, 2006, Guilin. **Proceedings [...]**. Guilin: IEEE, 2006. p. 1519.

ZHANG, D. D.; ZHANG, P. F.; WANG, L. S. Cell Counting Algorithm Based on YOLOv3 and Image Density Estimation. In: 4TH IEEE INTERNATIONAL CONFERENCE ON SIGNAL AND IMAGE PROCESSING (ICSIP), 2019, Wuxi. **Proceedings [...]**. Wuxi: IEEE, 2019. p. 920-924.

ZHANG, M.; LI, X.; XU, M. J.; LI, Q. Z. Automated Semantic Segmentation of Red Blood Cells for Sickle Cell Disease. **IEEE Journal of Biomedical and Health Informatics**, v. 24, n. 11, p. 3095-3102, 2020.

ZHAO, J. W.; ZHANG, M. S.; ZHOU, Z. H.; CHU, J. J.; CAO, F. L. Automatic detection and classification of leukocytes using convolutional neural networks. **Medical & Biological Engineering & Computing**, v. 55, n. 8, p. 1287-1301, 2017.

ZHENG, G. A.; LEE, S. A.; YANG, S.; YANG, C. H.; IEEE. Subpixel resolving optofluidic microscope based on super resolution algorithm. In: 8TH IEEE INTERNATIONAL SYMPOSIUM ON BIOMEDICAL IMAGING (ISBI) - FROM NANO TO MACRO, 2011, Chicago. **Proceedings [...]**. Chicago: IEEE, 2011. p. 1362-1365.

ZHONG, Z.; WANG, T.; ZENG, K.; ZHOU, X. G.; LI, Z. Y. White Blood Cell Segmentation via Sparsity and Geometry Constraints. **IEEE Access**, v. 7, p. 167593-167604, 2019.