

INSTITUTO FEDERAL DE SANTA CATARINA

ANTHONY CRUZ

APLICANDO DEEP LEARNING PARA DETECÇÃO DE CÉLULAS EM
AMOSTRAS DE SANGUE

Caçador - SC

06 de Dezembro de 2021

ANTHONY CRUZ

APLICANDO DEEP LEARNING PARA DETECÇÃO DE CÉLULAS EM AMOSTRAS DE SANGUE

Monografia apresentada ao Curso de Sistemas de Informação do Câmpus Caçador do Instituto Federal de Santa Catarina para a obtenção do diploma de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Samuel da Silva Feitosa

Coorientador: Prof. Me. Cristiano Mesquita Garcia

Caçador - SC

06 de Dezembro de 2021

Anthony Cruz

APLICANDO DEEP LEARNING PARA DETECÇÃO DE CÉLULAS EM AMOSTRAS DE SANGUE/ Anthony Cruz. – Caçador - SC, 06 de Dezembro de 2021-
46 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Samuel da Silva Feitosa

Monografia (Graduação) – Instituto Federal de Santa Catarina – IFSC
Campus Caçador
Sistemas de Informação, 06 de Dezembro de 2021.

1. Palavra-chave1. 2. Palavra-chave2. 2. Palavra-chave3. I. Orientador. II. Instituto Federal de Santa Catarina. III. Campus Caçador. IV. Título

ANTHONY CRUZ

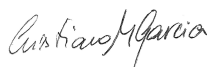
APLICANDO DEEP LEARNING PARA DETECÇÃO DE CÉLULAS EM AMOSTRAS DE SANGUE

Este trabalho foi julgado adequado para obtenção do título de Bacharel em Sistemas de Informação, pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

Caçador - SC, 06 de dezembro de 2021.



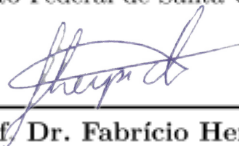
Prof. Dr. Samuel da Silva Feitosa
Orientador
Instituto Federal de Santa Catarina



Prof. Me. Cristiano Mesquita Garcia
Coorientador
Instituto Federal de Santa Catarina



Prof. Dr. Vitor Sales Dias da Rosa
Banca Avaliadora
Instituto Federal de Santa Catarina



Prof. Dr. Fabrício Herpich
Banca Avaliadora
Instituto Federal de Santa Catarina

*Dedico este trabalho a todos
que me apoiaram em
minha formação acadêmica.*

AGRADECIMENTOS

Tenho muito a agradecer, pois sozinho nenhuma grande obra jamais seria possível. Gostaria primeiramente de agradecer a Deus, por ter possibilitado essa incrível caminhada que participei em minha vida. Agradeço muito pela orientação recebida pelo Prof. Samuel, que me direcionou por toda a caminhada do TCC, mostrando os melhores caminhos a seguir e como atingir os melhores resultados. Agradeço também pela orientação do Prof. Cristiano que me ajudou a encontrar problemas de código e a sempre buscar realizar o melhor trabalho possível.

Sou muito grato pela presença da banca, ao Prof. Vitor que está participando mesmo distante atualmente do nosso campus, sempre foi de extrema importância para o meu desenvolvimento dentro da instituição e também fora dela. Também agradeço ao Prof. Fabrício pela disponibilidade e interesse em participar e dar todo o apoio desde o primeiro momento.

Agradeço à minha família e amigos que sempre estiveram ao meu lado nas minhas decisões. Agradeço também à minha namorada Josiani, que além de me dar forças e suporte em todas as etapas desse trabalho e também da vida, também muito contribuiu me explicando conceitos da área da saúde que eram desconhecidos por mim.

Agradeço ao IFSC, a instituição que por muitas vezes foi uma segunda casa para mim e contribuiu tanto no meu crescimento. À todos aqueles que fizeram parte e estavam presentes de alguma forma, minha sincera gratidão. Agradeço muito pela vida e por este momento tão incrível.

“Talk is cheap. Show me the code.”
(Linus Torvalds)

RESUMO

O hemograma é um dos exames mais importantes e mais realizados no âmbito médico. Através dele é possível descobrir importantes alterações no organismo e geralmente é utilizado como um primeiro passo na avaliação da saúde dos pacientes. Embora seja uma prática comum, a realização dos exames é dificultada nos laboratórios por utilizar um maquinário de alto custo de compra e manutenção. Como alternativa a isso, este projeto desenvolve um modelo computacional de *Deep Learning* capaz de detectar células automaticamente através de imagens de amostras de sangue. Através da utilização de bibliotecas de detecção de objetos, foi possível realizar o treinamento de um modelo voltado para essa problemática e capaz de detectar células em imagens com precisão satisfatória. Considerando a identificação de células em imagens de amostras de sangue nos melhores resultados obtidos, foi possível realizar a contagem das células brancas com 100% de acurácia, as vermelhas com 89% e das plaquetas com 96%, gerando subsídios para elaborar as partes principais de um hemograma. Os elementos voltados a classificação de diferentes tipos de células brancas não foram realizados devido à limitações do *dataset* utilizado. Porém, como apresentou bons resultados, a pesquisa pode ser expandida para futuros trabalhos que abordem esse problema.

Palavras-chave: Hemograma. Exames de Sangue. *Deep Learning*. Redes Neurais.

ABSTRACT

The complete blood count is one of the most important and most performed exams in the medical field. Through it, it is possible to discover important alterations in the organism and it is generally used as a first step in the assessment of patients health. Although it is a common practice, the performance of exams is difficult in laboratories because of the high cost of purchase and maintenance machinery. As an alternative to that, this project develops a computational model of *Deep Learning* capable of detecting cells automatically through images of blood samples. Through the use of object detection libraries, it was possible to train a model aimed at this problem and capable of detecting cells in images with satisfactory precision. Considering the identification of cells in images of blood samples in the best results obtained, it was possible to count white cells with 100% accuracy, red cells with 89% and platelets with 96%, generating subsidies to elaborate the main parts of a blood count. The elements aimed at classifying different types of white cells were not carried out due to the limitations of the *dataset* used. However, as it showed good results, the research can be expanded to future works that address this problem.

Keywords: Complete Blood Count. Blood Test. Deep Learning. Neural Networks.

LISTA DE ILUSTRAÇÕES

Figura 1 – Glóbulos Vermelhos (RBC)	18
Figura 2 – Glóbulos Brancos (WBC)	19
Figura 3 – Plaquetas (Platelets)	20
Figura 4 – Exemplo de Eritrograma e seus Atributos	21
Figura 5 – Exemplo de Leucograma e seus Atributos	21
Figura 6 – Matriz de Confusão	23
Figura 7 – <i>Perceptron</i>	24
Figura 8 – Artificial Neural Network	25
Figura 9 – Convolutional Neural Network	27
Figura 10 – Exemplo de aplicação de <i>Object Detection</i> no TensorFlow.	33
Figura 11 – Interface do protótipo desenvolvido.	37
Figura 12 – Detecção de Célula - WBC	40
Figura 13 – Detecção de Célula - WBC	40
Figura 14 – Detecção de Células - RBC	41
Figura 15 – Detecção de Células - RBC	41
Figura 16 – Detecção de Células - Platelets	42
Figura 17 – Detecção de Células - Platelets	42

LISTA DE TABELAS

Tabela 1 – Palavras-Chave e Sinônimos	29
Tabela 2 – Bases de Dados e Número de Artigos Seleccionados	29
Tabela 3 – Critérios de Exclusão	30
Tabela 4 – Artigos Seleccionados	31
Tabela 5 – Parâmetros Definidos	38
Tabela 6 – MSE e rMSE	39

LISTA DE ABREVIATURAS E SIGLAS

RBC Red Blood Cells

WBC White Blood Cells

CBC Complete Blood Count

VCN Volume Corpuscular Médio

HCM Hemoglobina Corpuscular Média

CHCM Concentração de Hemoglobina Corpuscular Média

RDW Red Cell Distribution Width

KNN K-Nearest Neighbors

ANN Artificial Neural Network

DNN Deep Neural Network

RNN Recurrent Neural Network

CNN Convolutional Neural Network

SVM Support Vector Machine

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Problema de Pesquisa	15
1.2	Hipótese de Pesquisa	15
1.3	Objetivos	15
1.3.1	Objetivo Geral	15
1.3.2	Objetivos Específicos	15
1.4	Justificativa	15
1.5	Organização do texto	16
1.6	Código-Fonte	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Exames Laboratoriais de Sangue	17
2.1.1	Sangue	17
2.1.1.1	Glóbulos Vermelhos	17
2.1.1.2	Glóbulos Brancos	18
2.1.1.3	Plaquetas	19
2.1.2	Hemograma	20
2.1.2.1	Eritrograma	20
2.1.2.2	Leucograma	21
2.2	Inteligência Artificial: <i>Machine Learning e Deep Learning</i>	21
2.2.1	Machine Learning	22
2.2.1.1	Algoritmos Tradicionais	23
2.2.1.2	Redes Neurais	24
2.2.2	Deep Learning	25
2.2.2.1	<i>Recurrent Neural Networks (RNNs)</i>	26
2.2.2.2	<i>Convolutional Neural Networks (CNNs)</i>	26
2.2.2.3	Bibliotecas e Recursos	27
3	ESTADO DA ARTE DA ÁREA PESQUISADA	29
3.1	Mapeamento Sistemático da Literatura	29
3.1.1	Critérios de Exclusão	30
3.1.2	Critérios de Inclusão	30
3.2	Análise dos trabalhos selecionados	30
4	METODOLOGIA	32
4.1	<i>Dataset</i>	32
4.2	Detecção de Objetos	33
4.3	Modelo Computacional	34
4.4	Avaliação	35
4.5	Protótipo	36
5	RESULTADOS	38
5.1	Análise de Desempenho do Modelo	38

5.2	Identificação de Células Brancas (WBC)	39
5.3	Identificação de Células Vermelhas (RBC)	40
5.4	Identificação de Plaquetas (<i>Platelets</i>)	41
5.5	Elaboração de Elementos do Hemograma	43
6	CONCLUSÕES	44
	REFERÊNCIAS	45

1 INTRODUÇÃO

A saúde humana sempre foi uma área pilar de toda a sociedade e vem se tornando ainda mais vital para sustentar as demais. Considerando os problemas e situações advindos da pandemia de COVID-19, é necessário pensar em formas de automatizar tarefas para agilizar o trabalho dos profissionais de saúde, permitindo que os mesmos tenham foco em problemas que demandam atenção urgente. Também, com o avanço da tecnologia e dos meios de comunicação, a automação vem se fazendo presente na vida de todos, tornando-se indispensável nas mais diversas áreas.

Também como efeito da pandemia, a demanda por exames laboratoriais vêm crescendo, e conforme isso acontece, um maior número de profissionais da saúde são necessários para atender, analisar e produzir laudos desses exames. Porém, nem todo local possui uma equipe adequada para lidar com este cenário, e então acontece sobrecarga de funções para dar conta dessa demanda.

Este trabalho tem como principal objetivo buscar maneiras de auxiliar a produção de hemogramas, através de detecções de imagens de sangue. Buscando criar uma ferramenta para auxiliar os profissionais da saúde nessa tarefa. Atualmente, os hemogramas são realizados por máquinas especializadas, portanto, demandam um alto custo financeiro e de manutenção para isso. Esse processo pode ser facilitado com o uso de algoritmos de *Deep Learning* para a automatização, como forma alternativa ao maquinário especializado.

Os algoritmos de *Deep Learning* (DL) vêm sendo utilizados nas mais diversas áreas, como na medicina (KRITTANAWONG et al., 2019), na economia (AKANBI et al., 2020), nas áreas da educação (OFFIR; LEV; BEZALEL, 2008), no comércio eletrônico (HA; PYO; KIM, 2016) e até em jogos virtuais (GREENGARD, 2017). Portanto, DL vem se tornando, cada vez mais, uma alternativa aos métodos tradicionais de realizar tarefas e automatizar processos. Podem ser encontrados alguns trabalhos também na área da saúde, que utilizam técnicas de *Deep Learning* como forma de auxiliar os profissionais em suas tomadas de decisão (RAVI et al., 2017; ZHAO et al., 2019).

As técnicas de *Deep Learning* buscam atingir resultados a partir de um grande conjunto de dados. Esses dados devem ser devidamente coletados e adaptados, ou seja, pré-processados adequadamente para que o processo de aprendizagem do *Deep Learning* possa ser desempenhado da melhor forma possível. O modelo de *Deep Learning* passa pela fase de treinamento e posteriormente por uma fase de realização de testes com outros dados para obtenção de resultados, que serão pós-processados para uma melhor visualização e apresentados ao profissional da saúde.

Nesse trabalho, busca-se analisar imagens de exames de sangue, para uma detecção e contagem automatizada das células da amostra. Normalmente, considerando o processo manual, essas amostras são verificadas e analisadas com o uso de lâminas de vidro e microscópio, onde são realizadas as devidas contagens de células, de forma a elaborar hemogramas a partir dessas informações. Outra abordagem utilizada em laboratórios de maior escala é a utilização de máquinas especializadas para esta contagem, que como já mencionado, envolvem custo financeiro elevado. Neste sentido, este projeto consiste na utilização de um *dataset*, grande conjunto de imagens de amostras de sangue, a fim de detectar e contar os diferentes tipos de células existentes na amostra, de forma a auxiliar os profissionais a elaborarem hemogramas.

1.1 Problema de Pesquisa

Considerando as diferentes formas e aplicações dos algoritmos de *Deep Learning* presentes nas mais diversas áreas, como seria possível utilizar um modelo computacional para a detecção e contagem de células em imagens de amostras de sangue, a fim de gerar informações úteis na elaboração de elementos do hemograma e auxiliar os profissionais de laboratório e da saúde?

1.2 Hipótese de Pesquisa

A hipótese para o problema apresentado é que modelos computacionais podem ser treinados para a detecção e contagem de células em imagens de amostras de sangue, com grande eficiência em prover informações úteis na elaboração de elementos do hemograma, para auxiliar os profissionais de laboratório e da saúde.

1.3 Objetivos

1.3.1 Objetivo Geral

Como objetivo geral deste trabalho, propõe-se o treinamento de um modelo computacional para detecção de células em imagens de amostras de sangue, voltado a prover informações úteis para elaborar elementos do hemograma e assim auxiliar os profissionais da área em suas tarefas.

1.3.2 Objetivos Específicos

- Realizar um mapeamento sistemático sobre o tema, a fim de reconhecer os trabalhos mais relevantes na literatura sobre as técnicas de *Deep Learning* utilizadas na área da saúde.
- Buscar imagens de amostras de sangue em bases de dados disponíveis e voltadas para esta finalidade.
- Realizar o pré-processamento dos dados coletados a fim de preparar o conjunto para o treinamento do modelo computacional.
- Desenvolver e treinar modelos computacionais de *Deep Learning* a fim de encontrar informações na análise de amostras de sangue;
- Desenvolver um protótipo funcional através do modelo computacional, para ser usado por profissionais da área.

1.4 Justificativa

Este estudo busca demonstrar uma forma alternativa de análise das amostras de sangue para elaboração de elementos do hemograma, portanto seu principal foco é auxiliar os profissionais da saúde. A contribuição desse estudo poderá ajudar os profissionais a serem mais rápidos em suas decisões sem perder a assertividade, de forma a aumentar a eficiência da análise de exames laboratoriais. Principalmente em momentos de crise, onde a área da saúde é bastante afetada, é necessário ter formas alternativas e associativas em tarefas cotidianas e de extrema importância para a continuidade dos trabalhos. Através dos resultados desse projeto, pesquisadores da área da computação e também da saúde, poderão ter uma noção desse contexto e de suas possibilidades, cultivando novas iniciativas de trabalho, de forma a auxiliar em novas pesquisas e aplicações.

Outra questão relevante, é em relação aos custos associados, devido ao fato de que o maquinário utilizado hoje para a análise desses exames demanda um alto custo para a sua compra e manutenção. Esse trabalho também possibilitará a realização da análise laboratorial reduzindo a necessidade de compra desses equipamentos, e consequentemente diminuindo também os custos envolvidos nesse processo.

Embora já existam estudos utilizando *Deep Learning* e também estudos utilizando esses conceitos na área da saúde, esse trabalho tem a ideia de associar a análise dos modelos de *Deep Learning* com a elaboração de elementos do hemograma de uma forma automatizada. Logo, se faz necessária a investigação dos conceitos desse trabalho para essa e futuras pesquisas.

1.5 Organização do texto

O restante desse trabalho está organizado da seguinte maneira: No [Capítulo 2](#) são apresentados os principais conceitos relacionados a *Deep Learning*, bem como as técnicas estudadas. No [Capítulo 3](#) são apresentados os resultados do mapeamento sistemático da literatura. No [Capítulo 4](#) será detalhada toda a metodologia do trabalho, assim como no [Capítulo 5](#) os resultados da pesquisa. Por fim, no [Capítulo 6](#) são apresentadas as considerações finais acerca deste trabalho.

1.6 Código-Fonte

Todo o código-fonte apresentado neste trabalho foi desenvolvido na linguagem Python, utilizando a biblioteca e os recursos do Tensorflow. Os códigos utilizados, assim como alguns processos internos do desenvolvimento foram omitidos e ficaram implícitos na escrita final para não se afastar do propósito principal, porém todos os detalhes e aprofundamentos em código do processo podem ser acessados no repositório do Github ([CRUZ, 2021](#)).

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados os principais conceitos de *Deep Learning*, assim como as técnicas e algoritmos aplicados para auxiliar os profissionais de laboratório a realizarem hemogramas de uma forma eficiente. Além disso, serão apresentados os dados coletados de uma análise de sangue que dão subsídio para a geração de elementos de hemogramas.

2.1 Exames Laboratoriais de Sangue

Os exames laboratoriais de sangue, principalmente se tratando de hemogramas, são um tipo de exame simples, porém de extrema importância para a saúde humana. Através desses exames pode-se descobrir diversas informações sobre o organismo da pessoa examinada, inclusive detectar doenças e problemas antecipadamente, por exemplo, para diagnosticar anemia, deficiências nutricionais, parasitas no sangue, doenças virais e autoimunes. Também é possível identificar infecções, doenças como leucemia, diagnosticar efeitos de medicamentos e também o efeito de vários tipos de estresses sobre o corpo (LOKWANI, 2013; LONGO, 2013).

Um exame pode ser solicitado por um médico, ou a partir do interesse do próprio paciente, e realizado em um laboratório de análises clínicas, que será responsável por realizar a coleta, encaminhar para a análise específica e retornar o resultado. Todo esse processo é custoso em tempo de espera e também financeiramente, pois o maquinário para esse tipo de atividade constitui preço expressivo para aquisição e manutenção.

2.1.1 Sangue

O sangue é um elemento do corpo humano, que circula em estado líquido através de todo o sistema circulatório do organismo, sendo de importância para o funcionamento correto das células através da entrada e saída de substâncias que podem modificar a sua composição (VIVAS, 2017).

Pode ser dividido em duas principais partes: o plasma (ou soro) e a parte celular. O plasma é a principal parte de transporte de substâncias pelo sistema, sendo este formado pela ingestão de água e alimentos. Também pode ser chamado de soro, sendo possível diferenciá-los pela presença ou não de anticoagulantes utilizados dependendo do tipo da análise buscada e da intenção da pesquisa (NAOUM, 2007).

A segunda parte do sangue, que será objeto de estudo para este trabalho, é a parte celular que contém todas as células presentes no sangue e se classificam como glóbulos vermelhos, glóbulos brancos e plaquetas. Geralmente observa-se a presença de eritrócitos, vários tipos e classes de leucócitos e as plaquetas como um todo, que serão abordados um a um posteriormente (LONGO, 2013).

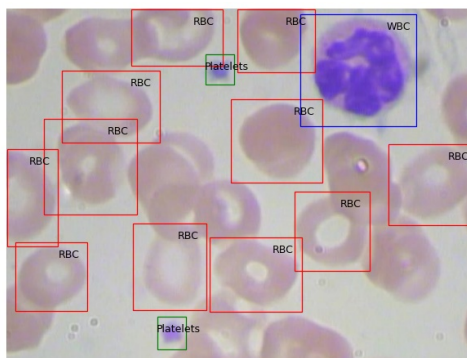
2.1.1.1 Glóbulos Vermelhos

Os glóbulos vermelhos, também conhecidos como *Red Blood Cells (RBC)*, são as hemácias presentes no sangue, também podem ser definidas em exames e registros médicos como eritrócitos. Essas células são pequenas e circulares, geralmente em formatos de discos e não possuem núcleo. Estão presentes em grande quantidade, possuindo uma vida útil de aproximadamente 120 dias até que o próprio sistema as elimine (LOKWANI, 2013).

É indispensável, ao falar sobre a parte vermelha do sangue, citar a hemoglobina que é uma proteína presente nas hemácias e de extrema importância para o funcionamento do sistema, pois através dela é possível realizar o transporte de oxigênio e gás carbônico pelo sistema sanguíneo, permitindo as trocas gasosas necessárias (VIVAS, 2017).

Percebe-se a presença dos glóbulos vermelhos na Figura 1, onde estão em grande quantidade em comparação com as outras células.

Figura 1 – Glóbulos Vermelhos (RBC)



Fonte: Shenggan (2019)

A Figura 1 faz parte de um *dataset*¹ específico de classificação de células sanguíneas, classificando em *RBC* (*Red Blood Cells*) para glóbulos vermelhos, *WBC* (*White Blood Cells*) para glóbulos brancos e *Platelets* para plaquetas.

2.1.1.2 Glóbulos Brancos

Os glóbulos brancos, também conhecidos como *White Blood Cells* (*WBC*), são as células brancas do sangue, sendo responsáveis pela defesa do organismo contra as principais ameaças do corpo humano presentes no sistema sanguíneo. Através da fagocitose, que é um processo de englobamento de partículas sólidas pelas células, são realizadas ações de defesa contra a invasão de fragmentos estranhos. Os glóbulos brancos são criados na medula óssea e estão presentes em todo o sangue, também em grande quantidade (LOKWANI, 2013).

Algumas células brancas podem ser encontradas na sua forma imatura, que ocorre quando o leucócito ainda não teve uma função definida pelo processo de maturação (promielócitos, mielócitos, metamielócitos). A maturação quando ocorre de forma adequada origina os seguintes tipos de leucócitos, cada qual com sua importância para o sistema de defesa do organismo (VIVAS, 2017):

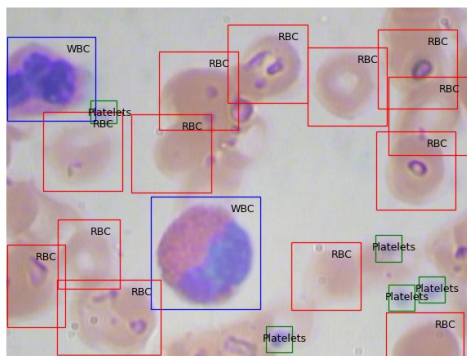
- **Neutrófilos:** células brancas mais abundantes, capazes de entrar nos tecidos, onde conseguem realizar a defesa do organismo, fagocitando partículas estranhas. Essas células são conhecidas como neutrófilos segmentados, pois existe uma célula precursora, que é o bastão, ou também chamado de neutrófilos bastonetes, que possuem essa nomenclatura pois seu núcleo não está amadurecido, ou seja ainda são jovens, e geralmente são identificados quando há infecções em fase aguda.
- **Eosinófilos:** células brancas responsáveis na defesa contra parasitas, geralmente estão presentes em grande quantidade no sangue durante reações alérgicas e infestações parasitárias.

¹ Um dataset é um grande conjunto de dados coletados.

- **Basófilos:** células brancas atuantes em respostas alérgicas e na coagulação do sangue. São capazes de liberar histamina, contribuindo para respostas alérgicas ao dilatar e permeabilizar os vasos sanguíneos e também liberam heparina que é capaz de prevenir a coagulação do sangue.
- **Monócitos:** células brancas capazes de entrar no tecido conjuntivo frouxo, onde conseguem se desenvolver em grandes células, com grande efeito fagocítico denominadas macrófagos, de forma a ingerir partículas estranhas ao organismo.
- **Linfócitos:** segundo tipo de célula branca mais abundante, são responsáveis e de extrema importância nas respostas imunes específicas do corpo humano, inclusive na produção de anticorpos.

Percebe-se também na Figura 2, a presença dos glóbulos brancos entre os vermelhos, porém em menor quantidade.

Figura 2 – Glóbulos Brancos (WBC)



Fonte: Shenggan (2019)

Como visto na Figura 2, embora existam várias classificações de glóbulos brancos, no *dataset* onde foi coletada, não era objetivo de estudo realizar essa análise específica de cada célula.

2.1.1.3 Plaquetas

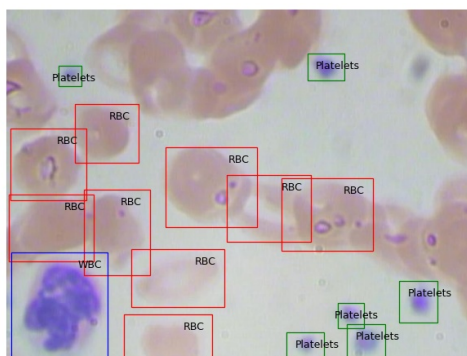
As plaquetas, também conhecidas e citadas como *Platelets*, são os menores componentes do sangue e possuem grande responsabilidade na hemostasia, que é uma resposta fisiológica para a prevenção e interrupção de sangramentos e hemorragias, ou seja, elas atuam na manutenção dos vasos sanguíneos. As plaquetas são fragmentos do citoplasma de megacariócitos, ou seja, elas são produzidas na medula óssea como parte dessas células especializadas que irão se dividir posteriormente e gerar um grande número de plaquetas. Aproximadamente, para cada 1 megacariócito, pode-se produzir cerca de 4000 plaquetas (LOKWANI, 2013).

Por serem fragmentos de uma célula, as plaquetas não possuem núcleo e são muito pequenas, com cerca de $1\text{--}3\ \mu\text{m}^1$ de diâmetro, com a coloração azul-acinzentado. A vida útil das plaquetas dura em média de 9 a 12 dias, e elas são removidas pelo baço quando estão velhas ou danificadas (LOKWANI, 2013).

Percebe-se também através da Figura 3, a presença do último elemento visível abordado por este trabalho no *dataset* utilizado como base. Nela é possível visualizar múltiplas plaquetas agrupadas em um dos cantos, além de outras no meio dos demais elementos do sangue.

¹ 1 micrômetro (μm) é equivalente à 0,001 milímetro.

Figura 3 – Plaquetas (Platelets)



Fonte: Shenggan (2019)

2.1.2 Hemograma

Um hemograma, também conhecido e citado como *Complete Blood Count (CBC)*, é um exame comum e muito utilizado, onde se realiza uma análise de sangue que envolve a contagem das diferentes células sanguíneas. A partir dos números obtidos através dessa contagem e com a comparação desse valor com as faixas de normalidade, é possível chegar a diversas conclusões sobre a saúde do paciente e até mesmo já identificar alguma doença ou problema (VIVAS, 2017; LOKWANI, 2013).

Um hemograma é geralmente realizado em duas principais etapas, sendo a primeira relacionada ao eritrograma que se refere à análise das células vermelhas, de forma a revelar até mesmo alguns tipos essenciais de alterações patológicas do sistema eritropoético, sendo o sistema responsável pela produção do material vermelho do sangue, como aumento na produção de glóbulos vermelhos e anemias. A segunda parte está relacionada com o leucograma, que corresponde à contagem global e específica dos leucócitos, a parte branca do sangue (VIVAS, 2017; LOKWANI, 2013).

2.1.2.1 Eritrograma

O objetivo do eritrograma ao realizar a análise da parte vermelha do sangue, é analisar alguns atributos-chave. Primeiramente é realizada a contagem geral dos eritrócitos adotando uma escala de milhões/mm³. A hemoglobina também é calculada e registrada em uma escala de g/dl (NAOUM, 2007; VIVAS, 2017).

Depois dessa contagem principal são calculados alguns índices, onde o primeiro deles apresenta o cálculo do volume corpuscular médio (VCM), sendo o volume médio das hemácias, calculado pelo quociente de um determinado volume de hemácias pelo número de células contidas no mesmo. Outro importante atributo é a hemoglobina corpuscular média (HCM), que semelhante ao VCM, apresenta o conteúdo médio da hemoglobina, calculado pelo quociente de conteúdo de hemoglobina em um determinado volume de hemácias pelo número de células contidas no mesmo volume (NAOUM, 2007; VIVAS, 2017).

Também temos outro índice que é a concentração de hemoglobina corpuscular média (CHCM), sendo a percentagem da hemoglobina em uma amostra de 100ml de hemácias. Por fim temos, a amplitude de distribuição dos glóbulos vermelhos, que em inglês significa *Red Cell Distribution Width (RDW)*, que será responsável por avaliar a variação de tamanho entre as hemácias (NAOUM, 2007; VIVAS, 2017).

É possível visualizar a forma que esses índices do eritrograma estão presentes e são abordados em um hemograma real através da Figura 4, assim como os seus respectivos valores de referência.

Figura 4 – Exemplo de Eritrograma e seus Atributos

Eritrograma		Valores de referência		
Eritrócitos.....	5,88	4,50 a	5,90	milhões/mm ³
Hemoglobina.....	16,6	13,5 a	17,5	g/dl
Volume Globular.....	49,4	41,0 a	53,0	%
VCM.....	84,0	80,0 a	100,0	fl
HCM.....	28,2	26,0 a	34,0	pg
CHCM.....	33,6	31,0 a	36,0	g/dl
RDW.....	13,7	11,5 a	15,0	%

Fonte: Elaborada pelo autor.

2.1.2.2 Leucograma

O objetivo do leucograma ao realizar a análise da parte branca do sangue, assim como no eritrograma, é analisar alguns atributos-chave, porém, diferente do processo anterior, essa etapa tem um foco maior na classificação e contagem de diferentes células brancas.

Primeiramente é feita uma contagem geral de leucócitos em mm³. Depois é realizada a contagem de forma a classificar cada tipo de leucócito presente, com neutrófilos, eosinófilos, basófilos, linfócitos, monócitos e também os granulócitos imaturos (promielócitos, mielócitos, metamielócitos). Por fim, também é calculado o número presente de plaquetas no sangue em mm³ (NAOUM, 2007; VIVAS, 2017).

É possível visualizar a maneira que os índices e classificações são abordados em um hemograma real através da Figura 5, assim como os seus respectivos valores de referência.

Figura 5 – Exemplo de Leucograma e seus Atributos

Leucograma		Valores de referência		
Leucócitos.....	7.520	/mm ³	3.500 a 10.000	/mm ³
Neutrófilos Bastonetes..	3 %	226 /mm ³	ate 840	/mm ³
Neutrófilos Segmentados..	43 %	3.234 /mm ³	1.700 a 8.000	/mm ³
Eosinófilos.....	8 %	602 /mm ³	50 a 500	/mm ³
Basófilos.....	1 %	75 /mm ³	até 100	/mm ³
Linfócitos.....	35 %	2.632 /mm ³	900 a 2.900	/mm ³
Linfócitos Atípicos.....	0 %	0 /mm ³		
Monócitos.....	10 %	752 /mm ³	300 a 900	/mm ³
Blastos.....	0 %	0 /mm ³		
Promielócitos.....	0 %	0 /mm ³		
Mielócitos.....	0 %	0 /mm ³		
Metamielócitos.....	0 %	0 /mm ³		
PLAQUETAS.....	282.000	/mm ³	150.000 a 450.000	/mm ³

Fonte: Elaborada pelo autor.

2.2 Inteligência Artificial: *Machine Learning* e *Deep Learning*

A abordagem de *Deep Learning*, que é objeto de estudo desse trabalho é uma das subáreas de *Machine Learning*, que, por sua vez, é subárea de Inteligência Artificial. Portanto antes de abordar cada um desses conceitos, quais suas diferenças e aplicações, se faz necessário introduzir o conceito de Inteligência Artificial.

A Inteligência Artificial é uma abordagem para a resolução de problemas de diversas naturezas de forma automatizada, ou seja, é uma maneira de se resolver problemas sem a necessidade de um humano ou usuário específico para esse trabalho. Essa área está recebendo muita atenção atualmente, buscando, cada vez mais, novas maneiras de automatizar tarefas do cotidiano. Essa abordagem está presente em

diversos segmentos da indústria e atualmente também vem crescendo o uso nas residências para as mais diversas finalidades (LIMA CARLOS A. M. PINHEIRO, 2014).

O intuito dessa área é buscar formas de ensinar máquinas e computadores a serem capazes de ter uma inteligência cada vez mais semelhante aos seres humanos. Isso ocorre geralmente através do reconhecimento de padrões através de diversas técnicas, que permitem que os computadores sejam ensinados a analisar e interpretar dados, de forma semelhante ao que acontece no aprendizado de seres humanos. Porém, diferente de nós, as máquinas geralmente precisam de um volume massivo de dados para que sejam capazes de aprender algo básico (FACELI et al., 2021).

2.2.1 Machine Learning

A resolução de problemas através de ferramentas da computação é comum e para isso se faz uso de algoritmos programados para finalidades específicas, porém não é em todos os problemas que se pode aplicar essa abordagem tradicional, porque nem sempre se sabe um caminho único de etapas a serem seguidas para chegar a uma resolução. Nesses casos, quando se sabe a entrada de dados e o resultado desejado, mas não os meios para se chegar nesse resultado, é possível utilizar um modelo de *Machine Learning* para realizar a predição. É possível pensar da seguinte forma, na abordagem de algoritmos tradicionais de programação, possuímos os parâmetros necessários e conhecemos o método para assim chegar ao resultado, mas na aplicação de *Machine Learning*, conhecemos os parâmetros e o resultado, porém o método será aprendido e apresentado pela própria máquina (ETHEM, 2016).

É possível fazer uma analogia entre um modelo de *Machine Learning* e uma criança que aprende algo novo, onde todo modelo passa por três principais etapas: o pré-processamento dos dados, a fase de treinamento, e por fim o teste. Na fase de pré-processamento, os dados são analisados e adaptados para um melhor entendimento do modelo, retirando informações inúteis, alterando o formato para um mais adequado, entre outras práticas (RASCHKA, 2015).

Logo após, o modelo de *Machine Learning* deve ser treinado conforme o método desejado, antes de realizar a predição de qualquer valor ou resultado, onde o responsável pelo treinamento deverá fornecer um grande volume de dados, assim como os resultados esperados em cada um deles, além dos parâmetros e configurações necessárias para o treinamento. Dessa forma o modelo aprende a entender e interpretar os dados, tornando-se ser capaz de reproduzir esse método para prever o resultado de futuros dados (RASCHKA, 2015).

Por fim, após o treinamento do modelo como um todo ocorre a fase de teste. Nesta fase, o modelo será testado e avaliado com base em algumas técnicas para medir o seu desempenho, calculando métricas como a acurácia, a precisão e a revocação. A acurácia indicará uma performance geral, ou seja, quantas classes o modelo classificou corretamente. A precisão dirá, dentre as classificações que o modelo classificou como positivo, quantas estão corretas. A revocação, também citada como *recall*, apontará dentre todas as classificações que possuem positivo como valor esperado, quantas estão corretas. Além dessas métricas, também existe o F1-Score que realiza uma média harmônica entre precisão e *recall* (NICK, 2018).

Para uma melhor visualização, e para realizar o cálculo das métricas citadas anteriormente, é comum a utilização de uma matriz de confusão, que descreve o número de acertos e erros relacionados com os verdadeiros ou falsos positivos e os verdadeiros ou falsos negativos, conforme observa-se na Figura 6 (NICK, 2018).

Figura 6 – Matriz de Confusão

		Classe Predita	
		Positivo	Negativo
Classe Verdadeira	Positivo	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Negativo	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Fonte: Elaborada pelo autor.

2.2.1.1 Algoritmos Tradicionais

Existem várias abordagens de *Machine Learning*, que são divididas em duas metodologias diferentes: supervisionados ou não-supervisionados. No aprendizado supervisionado, sabe-se o resultado correto, ou seja, onde o modelo deverá chegar a partir dos dados obtidos, porém no aprendizado não-supervisionado, o modelo terá que lidar com dados não estruturados e sem um resultado claro (RASCHKA, 2015).

Dentro do aprendizado supervisionado, encontra-se duas abordagens mais utilizadas, a regressão e a classificação. Os modelos de regressão serão responsáveis pela predição de valores reais, enquanto que os modelos de classificação serão responsáveis pela rotulação dos dados em determinadas classes (RASCHKA, 2015).

Quando se trabalha com regressão de dados, é possível citar alguns exemplos de abordagens mais utilizadas (NICK, 2018).

- **Linear Regression:** método mais clássico e simples para abordar regressão, onde é traçado uma reta através dos dados e então realizado a predição com base nessa reta. Possui variações como na *Multiple Linear Regression* onde são usadas mais variáveis e também na *Polynomial Linear Regression* onde a reta se tornará uma curva, utilizando variáveis com exponenciação, tornando o resultado mais assertivo.
- **Decision Tree:** árvore de decisão é uma técnica utilizada para prever valores através de critérios aprendidos pelo modelo. Esses critérios serão aprendidos através da divisão dos dados conhecidos em grupos semelhantes de forma a encontrar padrões em cada um dos grupos de dados.
- **Random Forest:** floresta aleatória é uma técnica utilizada de forma a combinar o poder de processamento de várias árvores de decisão diferentes, de forma a ter um trabalho mais minucioso e geralmente chegar em um resultado mais preciso.

Quando se trabalha com classificação de dados, é possível citar alguns exemplos de algoritmos mais utilizados (RASCHKA, 2015).

- **K-Nearest Neighbors (KNN):** o KNN é uma abordagem de um algoritmo que realizará a classificação com base na comparação de um dado com dados semelhantes a ele. É considerado uma

abordagem *lazy*, ou seja, preguiçosa, pois não apresenta um modelo inteligente, apenas armazena os dados e realiza comparações.

- **Decision Tree Classifier:** assim como na regressão, árvore de decisão é uma técnica utilizada para prever classes para dados específicos a partir de uma série de critérios aprendidos pelo modelo. Esses critérios serão aprendidos através da divisão dos dados conhecidos em grupos semelhantes de forma a encontrar padrões que nesse caso seriam as classes necessárias para a interpretação em cada um dos grupos divididos de dados.
- **Random Forest Classifier:** assim como na regressão, floresta aleatória é uma técnica utilizada de forma a combinar o poder de processamento de várias árvores de decisão voltadas à classificação, buscando ter um trabalho mais minucioso e geralmente chegar em um resultado mais preciso.

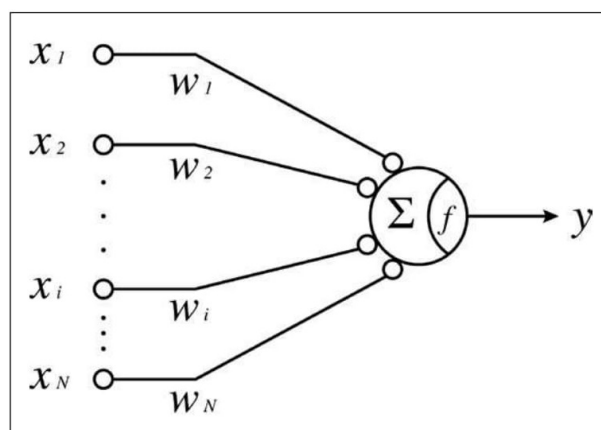
Além desses algoritmos, é importante ressaltar uma abordagem diferente da convencional que são as redes neurais, que podem ser utilizadas tanto para regressão quanto para classificação. Esse tipo de abordagem se difere dos algoritmos tradicionais e será objeto de estudo desse trabalho.

2.2.1.2 Redes Neurais

As redes neurais buscam uma abordagem parecida com o cérebro humano, pois utilizam uma estrutura composta de *perceptrons*, que são uma versão computadorizada de algo parecido com um neurônio humano. O *perceptron* é capaz de receber *inputs* (os dados de entrada) e produzir uma saída. O desempenho do uso de *perceptrons* pode ser melhorado ao utilizar vários em conjunto formando uma rede, assim unindo o poder de processamento de cada um deles e chegando a um resultado mais assertivo em menor tempo. Para essa estrutura se dá o nome de rede neural artificial ou como é conhecida, *Artificial Neural Network* (ANN) (SUGOMORI et al., 2017; GIANCARLO; REZAUL, 2018).

A Figura 7 apresenta toda a estrutura de um *perceptron*, onde o x representa a informação repassada, o w representa os pesos de cada informação, a letra Σ o somatório, o f a função de ativação e por fim o y representa a saída.

Figura 7 – *Perceptron*

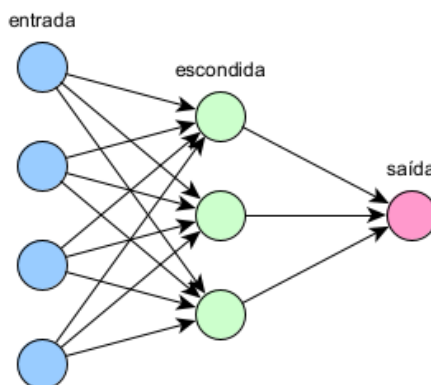


Fonte: Sugomori et al. (2017)

Uma *Artificial Neural Network* é uma estrutura formada por diversos neurônios, conhecidos como *perceptrons*, organizados de forma a imitar o processamento de um cérebro humano. Sua estrutura é definida em diversas camadas, geralmente adotando uma camada de entrada, outra de saída e camadas intermediárias chamadas de ocultas que serão responsáveis pelo processamento na totalidade, onde cada

uma delas pode conter um número variável de *perceptrons* (GIANCARLO; REZAUL, 2018). Pode-se visualizar esta estrutura na Figura 8, com a presença da camada de entrada, as camadas escondidas e por fim a camada de saída.

Figura 8 – Artificial Neural Network



Fonte: Elaborada pelo autor.

É importante ressaltar que além das informações repassadas de um *perceptron* para o outro, existe outro fator importante para o seu funcionamento, sendo os pesos adotados. Os pesos são valores definidos para cada informação recebida pelo *perceptron* e pode-se pensar no seu uso como um critério de prioridade entre uma informação e outra, que serão ajustados para alcançar o resultado desejado (GIANCARLO; REZAUL, 2018).

Dessa forma, uma ANN poderá aprender a realizar diversos tipos de tarefas, como classificação, por exemplo, através do uso de um algoritmo de *backpropagation* que significa propagação regressiva, responsável por gerir a taxa de aprendizado do modelo, seguindo alguns passos (GIANCARLO; REZAUL, 2018).

Primeiramente, o algoritmo inicia a rede neural artificial com pesos aleatórios, entrando em modo de treino para realizar o aprendizado. Durante o treinamento, o algoritmo é capaz de realizar previsões de resultados que serão comparados com os valores corretos, para saber se está obtendo sucesso ou não. O algoritmo de *backpropagation* então calcula a diferença do resultado obtido com o real, e essa informação do erro é repassada para todas as camadas anteriores de forma a realizar um ajuste nos pesos e minimizar o erro (GIANCARLO; REZAUL, 2018).

Todo esse processo é realizado muitas vezes durante o treinamento, e só encerra quando os ajustes levam ao aumento do erro novamente, indicando que os pesos foram ajustados até o limite. Essa etapa de otimização dos pesos utilizados pela rede é muito importante, sendo essencial para o funcionamento da rede neural, pois através dela que é possível para o modelo aprender com os seus erros (GIANCARLO; REZAUL, 2018).

2.2.2 Deep Learning

Existem alguns casos que o *Machine Learning* não é suficiente para o aprendizado a partir dos dados, pois o aprendizado acontece através de reconhecimento de padrões com base nos dados que não podem ser utilizados de qualquer forma, devem ser preparados e adaptados para cada modelo, ou seja, a máquina não é capaz de aprender por conta própria, pois precisa sempre de intervenção humana para o

processamento dos dados. Logo, é preciso uma abordagem mais parecida com a forma de pensar dos seres humanos, como o *Deep Learning* (ZOCCA et al., 2017; GIANCARLO; REZAUL, 2018).

O *Deep Learning* busca formas de aprofundar o conceito das redes neurais abordado na seção 2.2.1.2, utilizando *Deep Neural Networks (DNNs)*. As *Deep Neural Networks (DNNs)*, ou Redes Neurais Profundas, são uma arquitetura de rede neural orientada ao *Deep Learning*, ou seja, são mais complexas em seus modelos, com um número maior de neurônios, camadas ocultas em configurações mais complexas e conexões específicas entre os neurônios (GIANCARLO; REZAUL, 2018).

Com as possibilidades que as redes neurais profundas trouxeram, surgiram algumas variantes interessantes e focadas em um nicho específico, buscando se assemelhar ainda mais a natureza humana para realizar tarefas e criar inteligência. Algumas dessas variantes se aplicam a processamento de linguagem natural, como é o caso das RNNs abordadas na seção 2.2.2.1, ou também abordam o reconhecimento e interpretação de imagens como é o caso das CNNs abordadas na seção 2.2.2.2 e é assunto de interesse para a continuidade deste trabalho.

2.2.2.1 Recurrent Neural Networks (RNNs)

As *Recurrent Neural Networks*, ou Redes Neurais Recorrentes, são uma arquitetura de rede neural desenvolvida para a interpretação de dados temporais em sequência, ou seja, podem realizar previsões de variáveis em relação ao tempo. Sua estrutura é desenvolvida para permitir conexões de *feedback* das informações repassadas, ou seja, possui *loops* que permitem que as informações persistam, como se a rede fosse executada várias vezes. As RNNs são desenvolvidas para utilizar informações que possuem uma sequência fixa, como é o caso de informações que são atualizadas em decorrência do tempo (GIANCARLO; REZAUL, 2018).

2.2.2.2 Convolutional Neural Networks (CNNs)

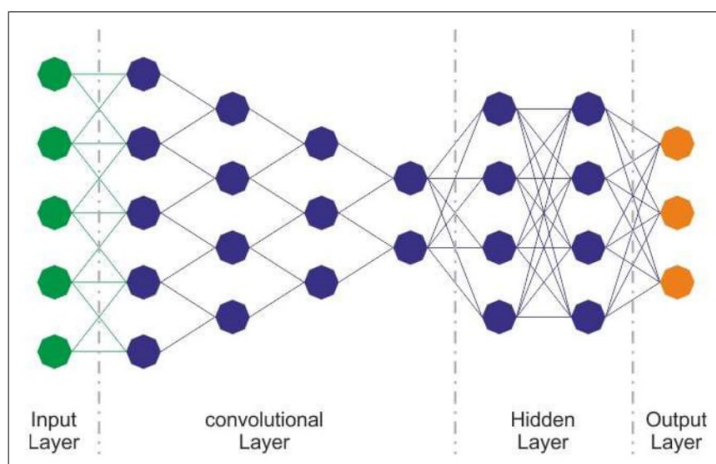
As *Convolutional Neural Networks*, ou Redes Neurais Convolucionais, são uma arquitetura de rede neural desenvolvida especificamente para o reconhecimento de imagens, com a capacidade de interpretar imagens dividindo-as em partes importantes. Essa rede trabalha inicialmente com 3 informações base, interpretando imagens como matrizes de 3 dimensões, sendo a altura, a largura e a cor (GIANCARLO; REZAUL, 2018).

As DNNs tradicionais funcionam bem para imagens pequenas, porém com um grande volume de dados, que nesse caso será cada pixel da imagem, o modelo teria muita dificuldade para aprender, logo nasceu a necessidade das CNNs. Para resolver esse problema, as CNNs foram desenvolvidas para usar camadas parcialmente conectadas e com grande reutilização dos pesos, dessa forma tem muito menos parâmetros à interpretar e passar adiante e consequentemente são mais rápidas para treinar, tendo menor risco de sobre-ajuste e necessitando de um volume menor de dados de treinamento (GIANCARLO; REZAUL, 2018).

Essa eficiência se deve também ao fato de que uma CNN, quando aprende a interpretar algum recurso ou elemento específico na imagem, torna-se capaz de identificar aquele elemento em qualquer lugar da imagem, diferentemente das DNNs que só aprendem a reconhecer um recurso em um local fixo. Isso mostra a capacidade das CNNs de serem mais generalistas em comparação com as DNNs (GIANCARLO; REZAUL, 2018).

Esse recurso pode ser encontrado dentro da camada convolucional, que está presente no início do processo e pode ser observada na Figura 9.

Figura 9 – Convolutional Neural Network



Fonte: [Giancarlo e Rezaul \(2018\)](#)

2.2.2.3 Bibliotecas e Recursos

Para se trabalhar com *Deep Learning* e suas principais arquiteturas e recursos, existem algumas bibliotecas disponíveis no mercado gratuitamente, as quais tendem a facilitar e orientar esse trabalho. Dessa forma, pode-se encontrar o Tensorflow e o Keras, sendo duas bibliotecas desenvolvidas em Python e C++, com a possibilidade de se trabalhar em conjunto e muito utilizadas pela comunidade de *Deep Learning* em geral ([GIANCARLO; REZAUL, 2018](#)).

O Tensorflow é uma plataforma completa de código aberto para *Machine Learning*, a qual considera uma grande variedade de tarefas com um foco nas redes neurais, porém suas funcionalidades também se estendem ao *Deep Learning*. As APIs disponibilizadas do Tensorflow utilizam como base o próprio Keras para definir e treinar as redes neurais. Desenvolvido pelo Google desde 2011, em 2018 a equipe decidiu integrar o Keras na biblioteca principal do Tensorflow ([TENSORFLOW, 2021](#)).

O Keras foi desenvolvido para permitir uma experimentação e prototipação rápida, com um principal foco nas redes neurais profundas. Promete ser fácil de utilizar, modular e ainda ser extensível. Possui vários recursos próprios, como a utilização de camadas, funções de perda, funções de ativação, entre outros. Além do foco em redes neurais profundas, possui módulos voltados a redes neurais convolucionais e recorrentes ([KERAS, 2021](#)).

O conjunto dessas duas bibliotecas para o trabalho com *Machine Learning* e *Deep Learning*, torna-se uma ferramenta extramente útil para este fim, possibilitando trabalhar com vários tipos de dados, como número, texto, áudio e também imagens.

Através da documentação do Tensorflow, é possível encontrar informações para desenvolver uma rede neural convolucional, por exemplo, onde existe toda a parte teórica e também a demonstração de um exemplo prático voltado a esse fim. Também é possível através dessa biblioteca e da sua documentação estar realizando a classificação de imagens com base em um *dataset* de imagens ([TENSORFLOW, 2021](#)).

Dentro do Tensorflow, também é possível estar trabalhando com diferentes módulos da biblioteca, nesse trabalho foi escolhido para uso o *Object Detection*. Esse módulo é utilizado dentro dessa área para um objetivo específico, que é a detecção de objetos contidos em uma imagem. Pode ser configurado para diversos usos, incluindo o caso desse trabalho, onde ao invés de detectar objetos físicos do dia a dia, o modelo é treinado para o reconhecimento de células em uma imagem e também para diferenciar uma

célula das outras.

Com o *Object Detection*, o tratamento das imagens e todo o processamento do modelo computacional se tornam tarefas mais fáceis de se trabalhar e também conta com outros vários recursos interessantes e úteis na hora do treinamento. Todo o algoritmo codificado para a realização do treinamento do modelo, contou com o apoio de funções do módulo do Tensorflow, *Object Detection*.

3 ESTADO DA ARTE DA ÁREA PESQUISADA

O processo de pesquisa e seleção dos trabalhos relacionados, foi realizado com base em um mapeamento sistemático sobre as pesquisas com propostas para agilizar a identificação e interpretação de hemogramas. Este mapeamento resultou na identificação e seleção dos principais trabalhos de pesquisa no tema deste Projeto de Trabalho de Conclusão de Curso. Outro objetivo deste mapeamento sistemático foi verificar os métodos utilizados para a aplicação de *Deep Learning* em imagens de amostras de sangue de maneira que possam ser obtidas informações para a geração automatizada de elementos de hemogramas.

3.1 Mapeamento Sistemático da Literatura

O mapeamento sistemático da literatura foi realizado com base na busca e levantamento de artigos, utilizando uma *string* de busca¹ nas principais bibliotecas e repositórios de artigos. Esses artigos foram analisados e selecionados conforme a sua área de pesquisa e a sua temática, para inclusão nesse estudo. Para gerenciar o mapeamento sistemático foi utilizado a ferramenta Parsifal², a qual auxilia na definição da *string* de busca, e permite salvar os artigos necessários e realizar a seleção.

As questões de pesquisas levantadas para isso foram, “Como os algoritmos de *Deep Learning* podem ser utilizados para a interpretação de exames?” e “Como realizar o tratamento de imagens para reconhecimento por modelos de *Deep Learning*?”. A partir dessas questões foram extraídas palavras e termos para o direcionamento da pesquisa. É possível visualizar estas palavras com seus sinônimos na Tabela 1.

Tabela 1 – Palavras-Chave e Sinônimos

Palavra-Chave	Sinônimos
Blood Analysis	Blood Sample
Classification	Interpretation, Recognition
Deep Learning	Artificial Intelligence, Computer Vision, Machine Learning

Fonte: Elaborada pelo autor.

Na Tabela 2, são listadas as bases de dados onde os artigos foram coletados, a quantidade de cada um deles e a *string* de busca utilizada na seleção. A mesma *string* de busca foi utilizada nas três bases de dados, e os artigos encontrados foram dos últimos 5 anos (2016-2021).

Tabela 2 – Bases de Dados e Número de Artigos Selecionados

Base de Dados	Artigos	String de Busca
ACM Digital Library	37	(“classification” OR “interpretation” OR “recognition”) AND (“deep learning” OR “artificial intelligence” OR “computer vision” OR “machine learning”) AND (“blood analysis” OR “blood sample”)
IEEE Digital Library	13	
Scopus	114	

Fonte: Elaborada pelo autor.

¹ Uma *string* de busca é um conjunto de palavras-chave organizadas para auxiliar na pesquisa.

² <https://parsif.al/>

3.1.1 Critérios de Exclusão

Os artigos coletados na pesquisa através da *string* de busca, passaram por critérios de exclusão por não se adequarem a esta pesquisa, esses critérios podem ser observados na Tabela 3.

Tabela 3 – Critérios de Exclusão

Critério de Exclusão	Nº de Artigos Recusados
O estudo não faz parte da área de pesquisa	101
O estudo apresenta resultados fora da computação	29
O estudo não é um estudo primário	6
O estudo é duplicado	16

Fonte: Elaborada pelo autor.

A seleção iniciou com 164 artigos no total das três bases de dados pesquisadas. Com a aplicação dos critérios de exclusão, observa-se um resultante de apenas 12 artigos. Isso ocorreu, pois, 101 artigos foram eliminados no critério “O estudo não faz parte da área de pesquisa”, que significa que esses artigos tinham alguma relação, porém eram voltados a outras áreas. Outros 29 artigos foram eliminados no critério “O estudo apresenta resultados fora da computação”, que significa serem da área de pesquisa, porém com resultados e métodos sem conexão com a computação. Foram também encontrados 6 artigos, que foram eliminados no critério “O estudo não é um estudo primário”, o que indica que o artigo pode ser uma revisão sistemática da literatura ou semelhante. Por fim, foram eliminados outros 16 artigos por serem duplicados.

3.1.2 Critérios de Inclusão

Os seguintes critérios de inclusão foram definidos da seguinte forma:

- Deve conter propostas de nova tecnologia para análise de hemogramas;
- Detalhar processo, método ou técnica para contagem de células sanguíneas;
- Desenvolver sistema para elaboração de hemogramas utilizando Deep Learning;

Na Tabela 4, é possível visualizar todos os 12 artigos selecionados com base nos critérios de inclusão, uma vez que todos eles se enquadram em pelo menos um deles.

Todos os artigos selecionados estão relacionados à recursos para auxiliar na interpretação de exames de sangue utilizando conceitos de *Deep Learning* e *Machine Learning*.

3.2 Análise dos trabalhos selecionados

Por fim, com os artigos selecionados e classificados, foi realizada a extração dos dados desses trabalhos, sendo essa a última etapa desse mapeamento sistemático da literatura. É possível perceber que os algoritmos e abordagens mais utilizados são técnicas de *Deep Learning*, como, por exemplo, o uso de *Convolutional Neural Network (CNN)* (A1, A2, A3, A4, A5, A6, A8, A9, A10, A11) e de *Recurrent Neural Network (RNN)* (A6, A11), sendo abordagens de redes neurais para a classificação das células sanguíneas.

Outros trabalhos utilizam de algoritmos de *Machine Learning* tradicionais para a classificação, como por exemplo, o uso de *Random Forest* ou *Decision Trees* (A2, A4, A7), sendo estruturas de árvores de decisão. Também foram encontrados estudos recorrendo a *Support Vector Machine (SVM)* (A7)

Tabela 4 – Artigos Seleccionados

ID	Título do Artigo	Autores
A1	Analyzing microscopic images of peripheral blood smear using deep learning	Mundhra, D. and Cheluvvaraju, B. and Rampure, J. and Rai Dastidar, T.
A2	Automatic detection and classification of leukocytes using convolutional neural networks	Zhao, J. and Zhang, M. and Zhou, Z. and Chu, J. and Cao, F.
A3	Automatic white blood cell classification using pre-trained deep learning models: ResNet and Inception	Habibzadeh, M. and Jannesari, M. and Rezaei, Z. and Baharvand, H. and Totonchi, M.
A4	Classification of Human White Blood Cells Using Machine Learning for Stain-Free Imaging Flow Cytometry	Lippeveld, M. and Knill, C. and Ladlow, E. and Fuller, A. and Michaelis, L.J. and Saeys, Y. and Filby, A. and Peralta, D.
A5	Blood cell classification using the hough transform and convolutional neural networks	Molina-Cabello, M.A. and López-Rubio, E. and Luque-Baena, R.M. and Rodríguez-Espinosa, M.J. and Thurnhofer-Hemsi, K.
A6	White Blood Cells Image Classification Using Deep Learning with Canonical Correlation Analysis	Patil, A.M. and Patil, M.D. and Birajdar, G.K.
A7	Image processing and machine learning in the morphological analysis of blood cells	Rodellar, J. and Alférez, S. and Acevedo, A. and Molina, A. and Merino, A.
A8	Improving blood cells classification in peripheral blood smears using enhanced incremental training	Al-qudah, R. and Suen, C.Y.
A9	Corruption-Robust Enhancement of Deep Neural Networks for Classification of Peripheral Blood Smear Images	Zhang, S. and Ni, Q. and Li, B. and Jiang, S. and Cai, W. and Chen, H. and Luo, L.
A10	Convolutional neural network and decision support in medical imaging: Case study of the recognition of blood cell subtypes	Diouf, D. and Seck, D. and Diop, M. and Ba, A.
A11	Combining Convolutional Neural Network With Recursive Neural Network for Blood Cell Image Classification	Liang, G. and Hong, H. and Xie, W. and Zheng, L.
A12	Blood diseases detection using classical machine learning algorithms	Alsheref, F.K. and Gomaa, W.H.

Fonte: Elaborada pelo autor.

que utilizam vetores de suporte e por fim *K-Means* e *K-Nearest Neighbors (KNN)* (A12), que fazem a classificação considerando os vizinhos mais próximos.

4 METODOLOGIA

Para a realização deste trabalho, muitos passos foram seguidos e desenvolvidos. Desde o estudo com o *dataset* escolhido, a preparação dos recursos necessários como as bibliotecas e o ambiente, até o treinamento do modelo e por fim o desenvolvimento do protótipo final. Neste capítulo, todas as etapas realizadas no projeto serão descritas em detalhes, para facilitar a compreensão dos passos executados para alcançar objetivos propostos.

4.1 Dataset

Para a realização de trabalhos na área de inteligência artificial, a escolha do *dataset* é uma tarefa de extrema importância e deve ser feita com o embasamento correto, pois impacta em todo o andamento dos processos seguintes. Durante a busca pelo *dataset*, alguns candidatos surgiram, mas a escolha que se mostrou mais adequada foi de utilizar o BCCD Dataset (SHENGGAN, 2019).

A escolha por esse *dataset* se deu, primeiramente, pelo maior número de trabalhos referenciando esta fonte. Também observou-se que em comparação com os outros *datasets* disponíveis na literatura, esse é o mais completo e aprofundado para esta área de pesquisa. Outro fator decisivo, foi a sua quantidade de imagens, que além de estarem em uma quantidade considerável para o uso, também apresentam boa qualidade para a interpretação visual e para o treinamento de modelos de *Deep Learning*.

O *dataset* apresenta três categorias de detecção WBC (*White Blood Cell*) para as células brancas, RBC (*Red Blood Cell*) para as células vermelhas e também as *Platelets*, para as plaquetas. Os arquivos estão separados em diretórios diferentes, onde em uma das pastas temos todo o conjunto de imagens, e em outra pasta temos as coordenadas de cada uma das células nas imagens, em formato XML.

Com o *dataset* em mãos, foi iniciado o estudo e aprofundamento de seu conteúdo, considerando os seus recursos e também suas limitações. Inicialmente foi possível perceber que uma das limitações do *dataset* é que nem todas as células nas imagens possuem coordenadas mapeadas. Grande parte apresenta o mapeamento correto, porém existem células que não apresentam nenhuma coordenada, e isso é uma falha do *dataset* que pode afetar o treinamento do modelo.

Outra limitação também observada nesse *dataset*, é a falta de classificação das células brancas. O *dataset* se preocupa apenas em localizar as células na imagem e categorizar em brancas, vermelhas e plaquetas. Porém, para se elaborar um hemograma completo, isso não é suficiente. Para a elaboração do hemograma, é necessário além de realizar a contagem das células brancas, também realizar a sua classificação (Neutrófilos, Basófilos, Eosinófilos, Linfócitos, Monócitos, entre outros), para através disso se ter um diagnóstico mais concreto a respeito do paciente. A introdução da detecção das classes para as células brancas ficará listada como possível trabalho futuro.

Com a obtenção do *dataset*, iniciou-se a fase de adaptação do conjunto de imagens para a detecção pelo modelo. No repositório, além das imagens, também estão contidos alguns programas *Python*, que servem para o tratamento dos dados como uma forma de demonstrar possíveis usos para o *dataset*. Como por exemplo, um programa capaz de fazer o contorno das células para uma pré-visualização dos dados. Porém, como este trabalho envolve a aplicação de um modelo computacional de *Deep Learning*, o tratamento dos dados será feito respeitando as necessidades desta técnica. Sendo assim, todos os arquivos *Python* contidos no *dataset* foram descartados para este trabalho, mantendo apenas o diretório das imagens e o diretório dos arquivos XML que contém as coordenadas das imagens.

4.2 Detecção de Objetos

Para o desenvolvimento do modelo computacional, assim como do treinamento e da configuração dos seus recursos, a biblioteca Tensorflow foi escolhida para atender a essa demanda. O Tensorflow possui um grande acervo de funções e utilitários dos mais diversos tipos para se trabalhar com *Machine Learning* e também *Deep Learning*, sendo necessário selecionar quais são os mais adequados a este trabalho. Um dos módulos mais interessantes e que se alinham com o intuito desse trabalho é o *Object Detection*, ou seja, o módulo de detecção de objetos ([TENSORFLOW, 2021](#)).

O módulo de detecção de objetos do Tensorflow é utilizado para encontrar elementos em porções de imagens, os quais foram previamente treinados para serem reconhecidos. Geralmente é utilizado para detectar pessoas e objetos do dia a dia em imagens. Também pode ser utilizado em vídeos, processando *frame a frame*, o que possibilita sua aplicação em câmeras de segurança, reconhecendo em tempo real pessoas, veículos, animais e outros elementos que aparecem com frequência. Um exemplo de detecção de objetos através do TensorFlow é apresentado na Figura 10.

Figura 10 – Exemplo de aplicação de *Object Detection* no TensorFlow.



Fonte: [Tensorflow \(2021\)](#)

Este trabalho consiste em criar um sistema de visão computacional capaz de detectar células em uma imagem, para assim realizar a contagem e posteriormente criar os elementos de um hemograma. Logo, o módulo de detecção de objetos se alinha com esse conceito. Embora seja normalmente utilizado para detecção de pessoas e objetos, este módulo possui grande capacidade de personalização, o que permite aprendizado voltado aos elementos que compõem o sangue. Ou seja, ao realizar o treinamento do modelo com base em um grande volume de imagens de sangue, é possível aprender com os dados para depois realizar as predições.

Como base para a utilização dessa biblioteca, a documentação foi de extrema importância para o entendimento dos seus recursos e limitações. No módulo de detecção de objetos, a documentação foi mais escassa, porém auxiliou em muitas questões do projeto, principalmente nos códigos base. Os primeiros códigos foram desenvolvidos com base nos exemplos de teste presentes na documentação, que permitiram ter uma grande noção das configurações possíveis de serem realizadas no modelo computacional e também de como é o funcionamento geral.

Para a execução dos códigos, porém, foi constatado que uma preparação do ambiente é necessária para atingir o melhor desempenho possível. Se fez necessário configurar corretamente a conexão do

algoritmo para execução na placa de vídeo (GPU), ao invés de executar no processador (CPU) como é o padrão. Como o foco é trabalhar com processamento de imagens, ainda mais com um grande volume, as GPU's possuem um desempenho mais eficaz, entregando resultados melhores em menor tempo.

Para a preparação do ambiente, foi necessário primeiramente realizar a instalação do CUDA. O CUDA é uma plataforma da NVIDIA para realização de computação paralela, para fornecer aos desenvolvedores o melhor aproveitamento do processamento das placas de vídeo e acelerar os aplicativos de computação. Adicional ao CUDA, também é interessante o uso de uma biblioteca conhecida por cuDNN. O cuDNN é uma biblioteca com foco em *Deep Neural Network*, ou seja, redes neurais profundas, que permite utilizar estruturas de redes neurais no trabalho com algoritmos de aprendizagem de máquina, sendo perfeitamente alinhado com o propósito desse trabalho. O uso do CUDA associado ao cuDNN, aumenta a eficiência e também a qualidade do resultado (CUDA, 2021; CUDNN, 2021).

Com a preparação do ambiente finalizada, o algoritmo já pode ser executado sem maiores problemas. Nessa etapa alguns testes foram realizados, executando os exemplos base presentes na documentação, para ter uma ideia de como o algoritmo se comporta na máquina e quais seriam os resultados do processamento de exemplo e em quanto tempo seriam entregues. Essa etapa foi importante para a continuidade dos trabalhos, pois foi possível ter uma noção de tempo necessário para execução dos códigos, permitindo aprimorar o planejamento das próximas etapas do projeto.

4.3 Modelo Computacional

Com todos os recursos, bibliotecas e ambiente funcionando em harmonia, iniciou-se o processo para inserir o *dataset* nos moldes do modelo para o treinamento e reconhecimento das imagens. O conjunto de imagens estava adequadamente formatado para o uso pelo algoritmo, estando em um tamanho suficiente, além de possuírem todas o mesmo tamanho e a mesma escala de ampliação.

Para o reconhecimento das coordenadas das imagens, primeiro, foi necessário a importação e tratamento dos arquivos XML em variáveis capazes de serem lidas pelo modelo. Contudo, apenas essa importação não foi o suficiente, visto que os dados contidos nos arquivos e os aceitos pelo modelo estavam em escalas e proporções diferentes. Ou seja, o algoritmo não estava conseguindo encontrar as células nas imagens durante a importação. Para adequar esse processo, foi necessário criar uma função capaz de converter as coordenadas do XML em números na escala do modelo, permitindo assim, importar tanto as coordenadas, quanto as imagens com sucesso.

Dessa forma, foi possível realizar um teste primário, sem alterar as características do modelo, para um dos tipos de células primeiramente, o que retornou resultados adequados para essa condição, com taxas baixas de acerto. Porém para se obter um ganho total dos conceitos do modelo, foi necessário a realização de ajustes nas configurações dos seguintes atributos:

- *Batch Size*: Significa o tamanho do lote utilizado no treinamento. Geralmente quanto maior o lote, melhor o aprendizado, porém também é necessário mais processamento da máquina.
- *Learning Rate*: Significa a taxa de aprendizado do modelo. Quanto maior, mais rápido os passos de aprendizado serão realizados. Caso seja muito grande, o modelo começa a reconhecer células onde não existem. Caso seja muito pequeno, o modelo não encontra nenhuma célula nas imagens.
- Número de *Batches*: Significa o número de passadas do algoritmo, também pode ser descrito como épocas. Define o intervalo de execuções que o algoritmo terá até parar o treinamento.
- Porcentagem de Acerto: Significa o número base para ser considerado um acerto de fato, por exemplo, se o algoritmo tem mais de 50% de certeza para uma célula, é considerado acerto.

Esses parâmetros foram configurados com base em testes realizados, onde a cada execução do código, foi verificado e ajustado conforme o necessário. Esse processo foi realizado até encontrar os valores mais adequados que retornam os melhores resultados. Os valores finais utilizados podem ser conferidos no capítulo 5.

Além da configuração desses parâmetros, também existe outro ponto importante e necessário para este trabalho, o qual consiste em selecionar uma rede neural pré-treinada adequada para o uso. Através de uma rede pré-treinada, muitos parâmetros comuns podem ser aproveitados, agilizando o processo de treinamento de novos modelos. Além disso, também é capaz de impactar nos resultados, pois se foi devidamente treinada com informações básicas, como contornos e elementos comuns, os resultados ao treinar com elementos mais complexos serão melhores.

Para este trabalho, foi selecionada a rede “resnet50”, embora existam muitas outras incluídas na lista da documentação aceitas pelo módulo de detecção de objetos. Essa escolha foi definida pelo tamanho da imagem utilizada e também ao realizar testes de desempenho em comparação com as outras redes (HE et al., 2015).

O uso da “resnet50” auxilia na fase de treino do modelo, pois não é necessário realizar o treinamento de toda a estrutura do zero e sim treinar apenas as últimas camadas. Como as camadas mais iniciais do modelo já foram treinadas com generalidades, toda a carga de trabalho está em treinar apenas as últimas camadas da rede com informações específicas. Nesse caso direcionar o aprendizado para imagens de amostras de sangue.

Com o treinamento de um tipo de célula realizado com sucesso, foi necessário expandir para treinar as demais. Primeiramente, foi testado treinando todas as células de uma só vez, porém essa abordagem estava consumindo muitos recursos da máquina e produzindo resultados insatisfatórios.

Nesse caso, a abordagem precisou ser alterada. Ao invés de treinar todos os tipos de células de uma só vez, foi criado um modelo único para cada uma delas, no caso um modelo para células brancas, outro para as vermelhas e um para as plaquetas. Isso permitiu que o treinamento fosse mais eficiente e com resultados mais assertivos e direcionados para cada tipo.

Como são utilizados 3 modelos, ao inserir uma imagem para verificação na predição de resultados, ela terá 3 respostas diferentes, sendo uma imagem de resultado para cada tipo de célula presente. Essa abordagem também permitiu utilizar valores de parâmetros diferentes para cada tipo, o que também auxiliou na melhora dos resultados para os diferentes tipos. Como o objetivo proposto neste trabalho é a contagem das células individuais, esta separação dos modelos para cada célula não tem influência negativa no resultado final.

4.4 Avaliação

Para a realização da avaliação, foram utilizadas métricas para medir o desempenho do modelo computacional e também para avaliar a qualidade dos resultados obtidos com cada um dos modelos treinados. As métricas foram utilizadas em cada uma das opções de modelos, utilizando configurações diferentes, para ter um comparativo entre uma opção e outra.

Dessa forma, foi possível avaliar e escolher com propriedade, qual algoritmo era o mais adequado para o uso no protótipo, assim como as melhores configurações possíveis para extrair o máximo desempenho da alternativa escolhida.

As métricas utilizadas para a avaliação, foram inicialmente a acurácia padrão para os primeiros testes. Posteriormente também foram medidas, utilizando o MSE (*Mean Squared Error*), que significa

erro quadrático médio e também foi estendido para o rMSE. o rMSE (*Root Mean Squared Error*), sendo uma variação do MSE, onde é aplicada a raiz quadrada ao resultado.

A acurácia padrão foi a primeira métrica utilizada, sendo calculada considerando o número de células reais para serem encontradas pelo modelo e o número de células que o modelo realmente conseguiu encontrar. Com base na relação desses números também foi calculada a porcentagem geral de acurácia final.

O MSE, foi calculado com base na fórmula apresentada a seguir. Primeiramente, foi calculada a diferença entre o valor encontrado pelo modelo y_i e o valor real \hat{y}_i para as n imagens, para então calcular o quadrado do resultado obtido. Por fim, é realizada a média desses valores para todas as imagens, gerando erro quadrático médio.

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

O valor ideal desse número é perto de 0, pois quanto maior o número, maior o erro do modelo. Como o resultado é sempre elevado ao quadrado, previsões distantes do real são facilmente perceptíveis, o que torna essa métrica interessante para problemas onde grandes erros não são tolerados.

O rMSE, foi calculado com base na fórmula apresentada abaixo. Como já foi calculado o MSE, foi possível calcular a raiz quadrada resultante para encontrar o rMSE final.

$$rMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}$$

Essa métrica é um complemento do MSE normal, pois através dessa métrica, é possível mensurar o resultado final sem alteração de escala do erro, facilitando algumas comparações. Pois, enquanto o MSE está elevado o quadrado, o rMSE não tem esse problema, pois é extraída a raiz quadrada no processo final.

4.5 Protótipo

Após a definição da melhor instância do modelo computacional possível, seguindo as métricas definidas, foi realizada a exportação desse modelo, para então utilizá-lo na criação de um protótipo funcional.

Para realizar a exportação do modelo, a documentação da biblioteca do Tensorflow precisou ser analisada novamente, estudando de forma mais aprofundada o funcionamento dos *checkpoints*. Os *checkpoints* são uma estrutura presente no Tensorflow que permitem salvar o estado atual de um modelo treinado para ser utilizado posteriormente. Dessa forma, é possível utilizar um modelo pronto, sem precisar passar pelo processamento do treinamento novamente e sim, somente utilizá-lo para previsões, com base em uma imagem de entrada. Sendo assim, o modelo pode ser usado em vários locais diferentes, apenas importando um arquivo (TENSORFLOW, 2021).

Para a criação do protótipo, foi utilizada uma biblioteca com foco em prototipação de algoritmos de inteligência artificial, o Streamlit. O Streamlit é um *framework* para o uso em Python, que permite criar protótipos de uma forma fácil e rápida. É possível ter uma aplicação web com poucos comandos, permitindo focar no mais importante que é o código. Logo, não é necessário se preocupar com a parte da interface gráfica (STREAMLIT, 2021).

Ainda assim é de fácil personalização, mostrando na tela todos os recursos do algoritmo, permitindo ao usuário enviar os dados de entrada (*input*) e também conseguir visualizar o resultado recebido do algoritmo (*output*). Dessa forma, é possível ter uma interação com o usuário em tempo real.

A Figura 11 apresenta a interface do protótipo desenvolvido.

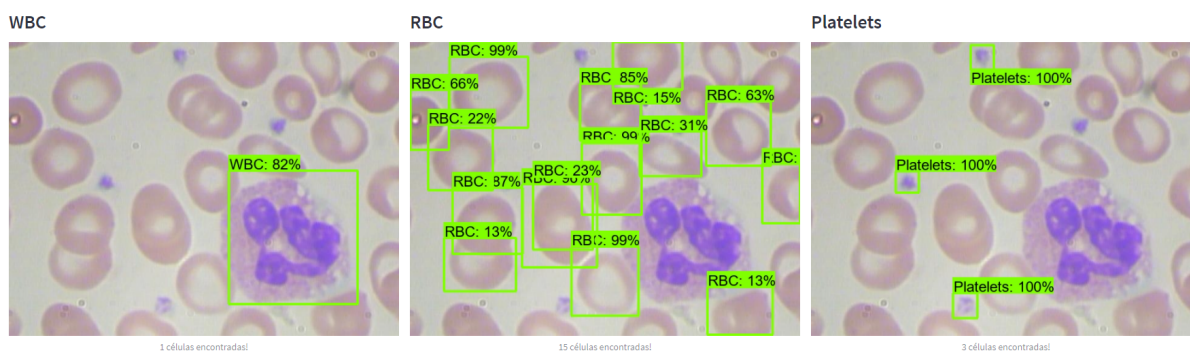
Figura 11 – Interface do protótipo desenvolvido.

Object Detection Aplicado em Células Sanguíneas

O modelo de Deep Learning foi treinado utilizando o BCCD Dataset, para fins de teste deve-se utilizar imagens dessa fonte.

Insira o link direto da imagem:

https://raw.githubusercontent.com/Shenggan/BCCD_Dataset/master/BCCD/JPEGImages/BloodImage_00355.jpg



Fonte: Elaborado pelo autor.

Como pode ser notado, o protótipo apresenta ao usuário um campo de entrada, onde é possível informar um *link* de uma imagem da Internet, sendo recomendado o uso de imagens no formato do *dataset* onde o modelo foi treinado. Na mesma Figura, também é possível visualizar um resultado do processamento de uma amostra de sangue, onde, nas imagens apresentadas, estão contidos o reconhecimento das células brancas, vermelhas e plaquetas, respectivamente, através no modelo treinado previamente. Além disso, também é possível identificar os percentuais de assertividade para cada detecção de célula realizada.

5 RESULTADOS

Neste capítulo serão abordados todos os resultados do projeto desenvolvido, os quais foram todos avaliados através das métricas e conceitos apresentados no Capítulo anterior. Dessa forma, será apresentado em detalhes como os resultados foram obtidos, medidos e avaliados a partir das métricas, e também apresentadas imagens produzidas pelo modelo treinado.

5.1 Análise de Desempenho do Modelo

O *dataset* utilizado como base para a realização deste trabalho, possui 364 imagens diferentes de amostras de sangue. Cada imagem pode conter até 30 células de classes diferentes, localizadas em diferentes porções da imagem. As classes utilizadas nessa pesquisa são células vermelhas (RBC), células brancas (WBC) e também plaquetas (*Platelets*), as quais são identificadas através de coordenadas que indicam a localização de cada célula na imagem, e ficam armazenadas em arquivos XML que acompanham o *dataset* em questão.

Este *dataset* foi separado em dois principais grupos. O primeiro grupo se refere ao conjunto de imagens voltados apenas para o treino, com 80% das imagens totais, esse grupo contém 291 imagens selecionadas aleatoriamente. Enquanto o segundo grupo, voltado apenas para o teste, com 20% das imagens totais, contém 73 imagens separadas para este fim. Como o grupo de treino precisa ser maior para se ter um bom treinamento do modelo e por fim realizar o teste nas imagens restantes, a seleção de 80/20 da separação dos arquivos foi adotada.

Como observado no Capítulo 4, o modelo foi configurado seguindo alguns princípios e configurações principais. Primeiramente foi selecionada a rede neural pré-treinada com o melhor resultado e performance para este grupo de imagem, sendo essa a “resnet50”, reconhecida na base de código por “ssd_resnet50_v1_fpn_640x640_coco17_tpu-8”. Além dessa escolha, quatro parâmetros fundamentais foram cuidadosamente selecionados, buscando o melhor desempenho em menor tempo. A definição dos parâmetros de *Batch Size*, *Learning Rate*, Número de *Batches* e também a porcentagem de acerto base podem ser visualizados na Tabela 5.

Tabela 5 – Parâmetros Definidos

	WBC	RBC	Platelets
Batch Size	24	24	24
Learning Rate	0.02	0.05	0.05
Número de Batches	500	500	500
Porcentagem de Acerto	0.5	0.1	0.5

Fonte: Elaborada pelo autor.

Após o treinamento do modelo, as predições foram medidas utilizando 3 métricas diferentes. A primeira foi a acurácia padrão, que permite ter uma visão geral do andamento do processo. Realizando a medição da acurácia padrão, considerando o número de células reais e o número de células que deveriam ser encontradas pelo modelo, se chegou na seguinte relação, utilizando o melhor resultado possível como parâmetro:

- WBC: 74 células encontradas de 74 células reais (100,00% de Acerto e 0,00% de Falha)

- RBC: 1007 células encontradas de 908 células reais (89,10% de Acerto e 10,90% de Falha)
- Platelets: 57 células encontradas de 59 células reais (96,61% de Acerto e 3,39% de Falha)

Conforme constatado no Capítulo 4, o MSE e o rMSE, são métricas complementares, de fácil utilização e se referem ao desempenho do modelo computacional. Enquanto o MSE trabalha realizando o cálculo do erro quadrático médio, o rMSE é a raiz desse valor encontrado. Tais métricas são interessantes de se utilizar, pois, permitem visualizar de forma considerável, até mesmo pequenos erros. Essa escala não apresenta um número máximo ou ideal, mas o resultado deve se aproximar de zero, quanto mais próximo, melhor. Dessa forma, utilizando as fórmulas do capítulo anterior, foi possível chegar na relação da tabela 6.

Tabela 6 – MSE e rMSE

	MSE	rMSE
WBC	0.3835616438356164	0.6193235372853323
RBC	14.342465753424657	3.787144802278447
Platelets	0.9041095890410958	0.9508467747440151

Fonte: Elaborada pelo autor.

Através da relação da Tabela 6, foi possível perceber que o melhor desempenho está na detecção de células brancas, o que se alinha com as definições encontradas pela acurácia vista anteriormente. Isso pode ter acontecido, pois, no *dataset*, em grande parte das imagens existe pelo menos uma célula branca que é facilmente detectada em relação às demais. Seguindo a isso, as plaquetas também apresentam bons resultados, parecido com a relação da acurácia do modelo. Por fim, as células vermelhas apresentam resultados inferiores em relação às demais. Isso pode ter acontecido, devido ao fato da sua abundância em todas as imagens, enquanto as outras classes apresentam uma ou duas células por imagem, as células vermelhas podem aparecer em um número maior que 20, isso dificulta o aprendizado do modelo e também atrapalha na predição nesse caso.

5.2 Identificação de Células Brancas (WBC)

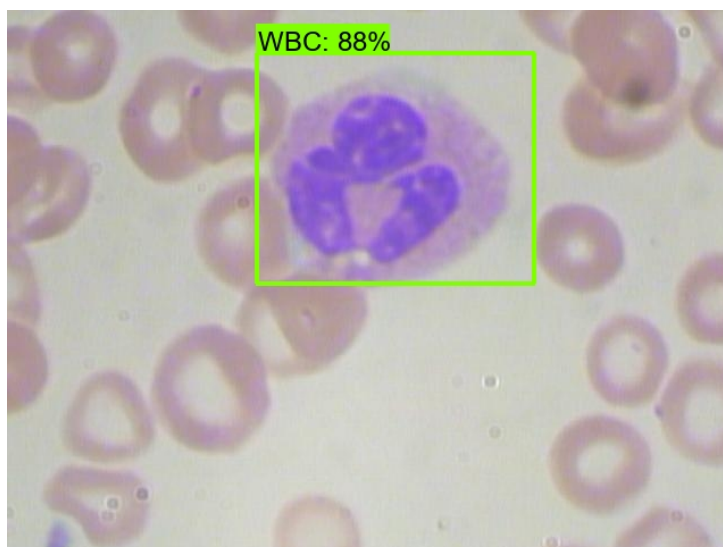
A primeira classe de células trabalhadas, desde o primeiro teste utilizando o *dataset* foram as células brancas. Isso ocorreu pela sua fácil identificação em relação às demais, sendo um ótimo candidato para se iniciar os testes e posteriormente as primeiras versões do protótipo.

Essas células estão presentes nas imagens em baixa quantidade, pois o *dataset* buscou focalizar uma célula branca em cada imagem. Embora em alguns casos seja possível encontrar até duas ou três. Além da quantidade, podem ser diferenciadas das outras através da sua coloração.

Na Figura 12 pode ser observado a detecção da célula branca pelo modelo. Onde foi possível diferenciar e delimitar a célula de forma satisfatória. Na Figura 13 também pode ser observado o mesmo processo, mas em outra situação.

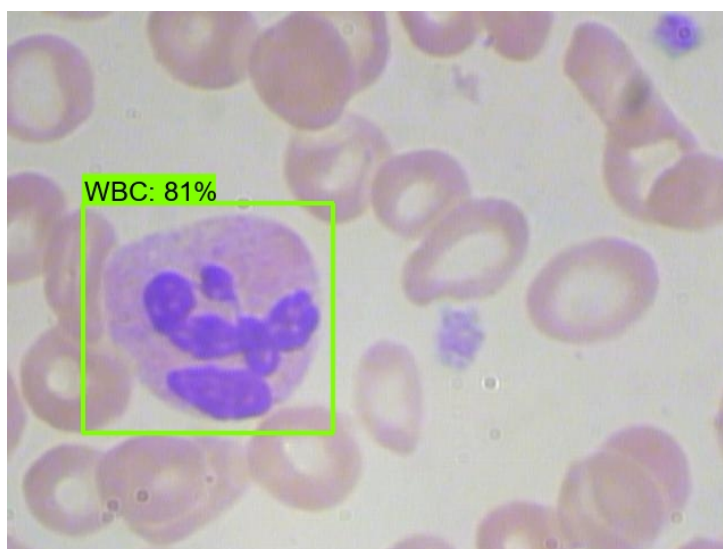
Pôde-se perceber que o modelo obteve boa taxa de acerto, levando em consideração tanto as métricas de desempenho, quanto através da análise visual das imagens. Deste modo, é possível realizar as contagens das células com bom índice de assertividade.

Figura 12 – Detecção de Célula - WBC



Fonte: Elaborado pelo autor.

Figura 13 – Detecção de Célula - WBC



Fonte: Elaborado pelo autor.

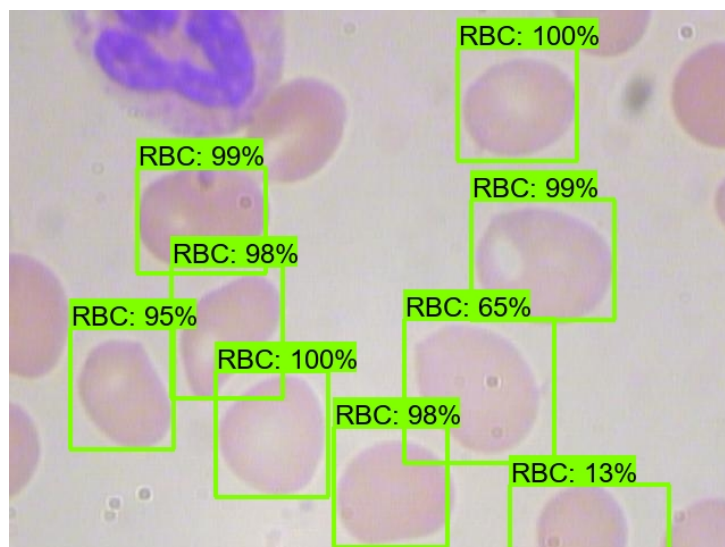
5.3 Identificação de Células Vermelhas (RBC)

Após a detecção das células brancas, o próximo passo buscado foi a interpretação das células vermelhas. Essas se diferenciam na sua detecção em relação às outras devido a sua quantidade. As células vermelhas aparecem em uma proporção maior, constando até mesmo mais de 20 células por imagem.

A grande quantidade desse tipo de células é um desafio para o modelo, principalmente pela delimitação de suas áreas. Por não terem forma definida e muitas vezes se sobrepondo uma sobre as outras, isso atrapalha a sua identificação.

Na Figura 14 podemos visualizar o processo de detecção das células vermelhas. Sendo possível realizar a contagem das células conforme o necessário, com uma pequena margem de erro.

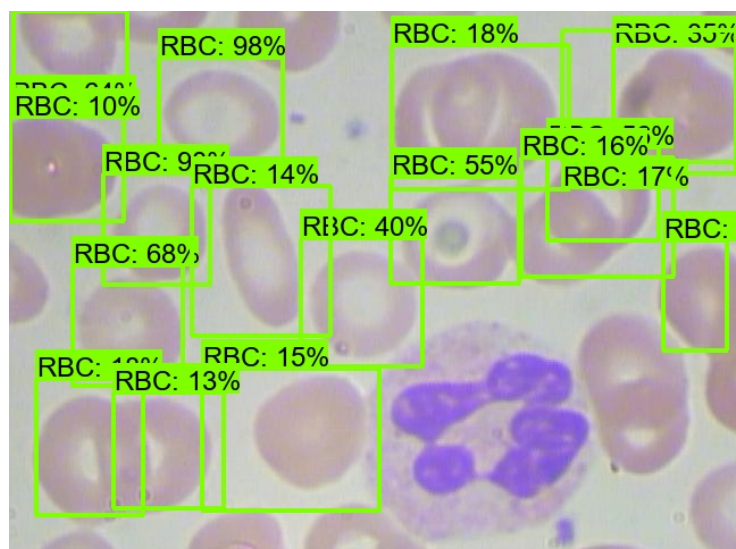
Figura 14 – Detecção de Células - RBC



Fonte: Elaborado pelo autor.

Entretanto, na Figura 15 também é possível visualizar a identificação pelo modelo, porém em uma densidade maior, mesmo com a sobreposição de células, ainda assim o modelo conseguiu identificar a maior parte das células de forma correta.

Figura 15 – Detecção de Células - RBC



Fonte: Elaborado pelo autor.

É possível concluir que embora não tenha um desempenho tão positivo quanto as demais classes, os resultados também são bons pois permitem uma contagem das células vermelhas com uma pequena margem de erro.

5.4 Identificação de Plaquetas (*Platelets*)

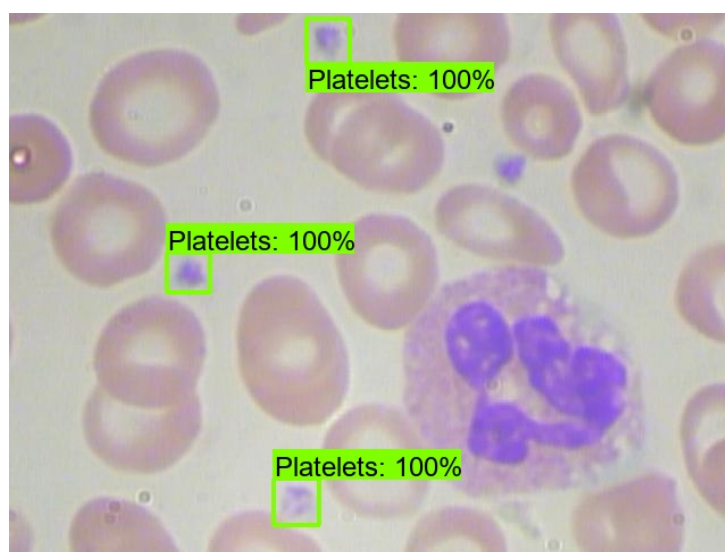
Por fim, a detecção das plaquetas foram realizadas de forma similar as células brancas. Apesar de estarem mais presentes no *dataset* do que as células brancas, elas não estão em um número tão elevado

quanto as células vermelhas, apresentando pequenos conjuntos por imagens. Outro fator característico em relação as demais é o seu tamanho, que é muito menor e de fácil identificação a olho nu.

Embora sejam identificadas de forma fácil, a sobreposição de células também pode acontecer nesse caso, atrapalhando o desempenho da identificação pelo modelo. Porém isso acontece em um número significativamente menor em comparação com as células vermelhas.

É possível visualizar na Figura 16 o processo de identificação de plaquetas, onde o modelo apresenta um desempenho considerado ótimo.

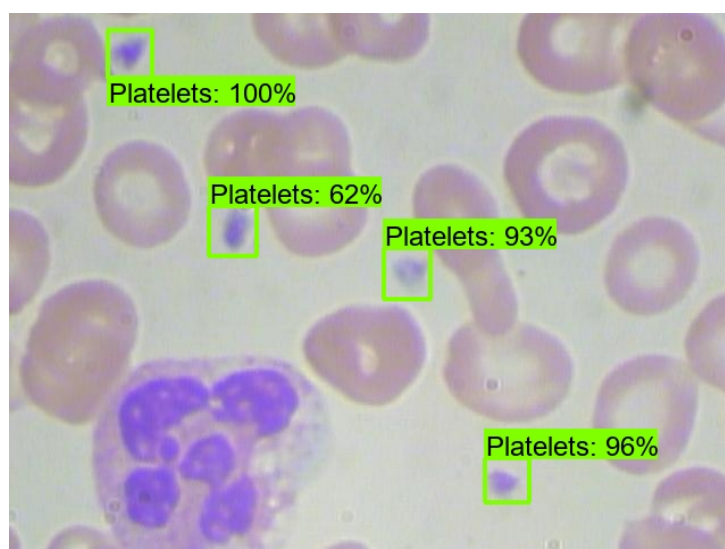
Figura 16 – Detecção de Células - Platelets



Fonte: Elaborado pelo autor.

Já na Figura 17 é possível notar uma situação um pouco diferente. Nessa análise o modelo foi capaz de identificar corretamente as 4 plaquetas que aparecem na imagem, porém com margem de assertividade do algoritmo levemente reduzida.

Figura 17 – Detecção de Células - Platelets



Fonte: Elaborado pelo autor.

Com isso, podemos concluir que as plaquetas possuem um desempenho adequado na sua identificação permitindo uma contagem satisfatória. Embora não seja tão positivo quando na detecção das células brancas, ainda supera as células vermelhas em relação ao desempenho e critérios de assertividade.

5.5 Elaboração de Elementos do Hemograma

Com base na identificação e na contagem das células, como visto anteriormente, é possível realizar a produção de elementos do hemograma. Um hemograma é basicamente uma contagem das células e comparação desses resultados com índices pré-calculados de valores ideais para cada tipo de paciente.

Os valores iniciais do hemograma podem ser calculados realizando, após a identificação das células, a contagem de cada tipo. As unidades de medida são importantes nesse processo, pois são diferentes para cada tipo de célula. A unidade base para o cálculo é o milímetro cúbico (mm^3), ou seja, uma unidade de volume. Para o cálculo dos leucócitos, que são as células brancas, e também das plaquetas é realizada a contagem em proporção a um milímetro cúbico. Porém, no caso dos eritrócitos, que são as células vermelhas, elas estão em uma quantidade maior em relação às outras células, portanto é realizado o cálculo de milhões de células em um milímetro cúbico.

O eritrograma pode ser elaborado, considerando o número de eritrócitos em uma relação de milhões/ mm^3 . O leucograma pode ser elaborado, utilizando o número de leucócitos encontrados em unidade/ mm^3 . E por fim o plaquetograma pode ser elaborado, com base no número de plaquetas identificadas em unidades/ mm^3 .

Essas informações são necessárias para detectar várias doenças e desequilíbrios do organismo do paciente. Como por exemplo, o leucograma sendo o principal exame para avaliar a imunidade de uma pessoa, através da quantidade de leucócitos presentes no organismo. O número das plaquetas também muito diz sobre o organismo, avaliando a coagulação sanguínea do indivíduo e pré-disposições à hemorragias. Assim como, o número de hemácias baixo, poderia indicar uma anemia ou doença associada, entre muitos outros casos.

Podemos concluir que através desses dados, é possível elaborar importantes elementos do hemograma que sendo aplicados em uma amostra com valores de volume, será útil e interessante para o uso. Em situações em que os valores encontrados apresentem resultados alterados, é necessário realizar uma investigação aprofundada com exames específicos a cada situação.

As limitações para a elaboração de um hemograma total estão relacionadas com as limitações do *dataset*. Primeiramente, para a realização de um hemograma completo, seria necessário ter além da contagem, também a classificação das células brancas. Sua classificação em Monócitos, Leucócitos, Neutrófilos, Eosinófilos, Basófilos, Linfócitos, entre outros é necessária para a completude do exame. Porém, essas classificações não são apresentadas pelo *dataset*, que procura focar apenas nas diferenciações entre os três principais tipos.

Outros valores relacionados à cálculos com volume, como o VCM (Volume Corpuscular Médio) e também o HCM (Hemoglobina Corpuscular Média), entre os outros citados no Capítulo 2 não são possíveis de serem calculados por falta de informação. Na utilização de outro *dataset* em trabalhos futuros, essas informações podem ser calculadas a partir da coleta de dados, como será abordado no próximo capítulo.

6 CONCLUSÕES

Este projeto iniciou com o interesse de utilizar as técnicas de *Deep Learning* para auxiliar os profissionais da saúde em seus trabalhos, pois com a automatização que essa técnica oferece, outras tarefas podem receber maior atenção. Dentre os diferentes tipos de exames realizados diariamente, o hemograma é o essencial, de forma a ser um ponto de partida para a identificação da maioria das doenças. Portanto, os estudos subsequentes foram direcionados a estudar quais seriam as melhores abordagens para a automatização dessa tarefa.

Foi possível entender todo o processo de produção de hemograma e também encontrar as formas de associar os conceitos de *Deep Learning* nesse processo. Através do treinamento de um modelo computacional inteligente capaz de reconhecer as diferentes células do sangue, desenvolveu-se a contagem de todos os tipos de célula com grande precisão.

Essa abordagem desenvolvida é de grande utilidade, pois permite tirar conclusões sobre o estado atual da saúde do paciente. A contagem das células brancas, vermelhas e também das plaquetas, permitem avaliar se existe algum desequilíbrio em seus níveis, caso apresente algum fator anormal, outros exames mais aprofundados podem ser realizados para verificar estas questões. Entretanto, não se caracteriza como um hemograma completo, pois devido a limitações do *dataset* utilizado, o trabalho também apresentou limitações, principalmente relacionado a classificação das células brancas, onde a contagem foi geral.

Para estudos futuros, é possível utilizar os conceitos aqui abordados para aplicar estes conhecimentos em um conjunto de dados diferente e mais complexo, ou também voltado para outro fim. Outra possibilidade é utilizar métricas de avaliação mais voltadas a essa prática, considerando a área das predições e intersecções com os valores reais. A partir deste resultado, é possível continuar a abordagem adotada por este trabalho para estimar a contagem em relação ao volume e, posteriormente, reproduzir todas as informações presentes em um hemograma.

REFERÊNCIAS

- AKANBI, L. A. et al. Deep learning model for demolition waste prediction in a circular economy. *Journal of Cleaner Production*, v. 274, p. 122843, 2020. ISSN 0959-6526. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0959652620328882>>. Citado na página 14.
- CRUZ, A. *blood-detection*. 2021. <<https://github.com/AnthonyLeier/blood-detection>>. Citado na página 16.
- CUDA. 2021. <<https://developer.nvidia.com/cuda-zone>>. Acessado em: 2021-10-31. Citado na página 34.
- CUDNN. 2021. <<https://developer.nvidia.com/cudnn>>. Acessado em: 2021-10-31. Citado na página 34.
- ETHEM, A. *Machine Learning : The New AI*. The MIT Press, 2016. (MIT Press Essential Knowledge Series). ISBN 9780262529518. Disponível em: <<http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1369420&lang=pt-br&site=ehost-live>>. Citado na página 22.
- FACELI, K. et al. *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina*. Rio de Janeiro: Livros Técnicos e Científicos Editora Ltda, 2021. Citado na página 22.
- GIANCARLO, Z.; REZAUL, K. M. *Deep Learning with TensorFlow : Explore Neural Networks and Build Intelligent Systems with Python, 2nd Edition*. Packt Publishing, 2018. v. 2nd ed. ISBN 9781788831109. Disponível em: <<http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1789473&lang=pt-br&site=ehost-live>>. Citado 4 vezes nas páginas 24, 25, 26 e 27.
- GREENGARD, S. Gaming machine learning. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 60, n. 12, p. 14–16, nov. 2017. ISSN 0001-0782. Disponível em: <<https://doi.org/10.1145/3148817>>. Citado na página 14.
- HA, J.-W.; PYO, H.; KIM, J. Large-scale item categorization in e-commerce using multiple recurrent neural networks. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2016. (KDD '16), p. 107–115. ISBN 9781450342322. Disponível em: <<https://doi.org/10.1145/2939672.2939678>>. Citado na página 14.
- HE, K. et al. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. Disponível em: <<http://arxiv.org/abs/1512.03385>>. Citado na página 35.
- KERAS. 2021. <<https://keras.io/>>. Acessado em: 2021-06-04. Citado na página 27.
- KRITTANAWONG, C. et al. Deep learning for cardiovascular medicine: a practical primer. *European Heart Journal*, v. 40, n. 25, p. 2058–2073, 02 2019. ISSN 0195-668X. Disponível em: <<https://doi.org/10.1093/eurheartj/ehz056>>. Citado na página 14.
- LIMA CARLOS A. M. PINHEIRO, F. A. O. S. I. *Inteligência Artificial*. Rio de Janeiro: Elsevier Editora Ltda, 2014. Citado na página 22.
- LOKWANI, D. *The ABC of CBC: Interpretation of complete blood count and histograms*. [S.l.]: Jaypee, 2013. Citado 4 vezes nas páginas 17, 18, 19 e 20.
- LONGO, D. L. *Atlas de Hematologia e Análise de Esfregaços do Sangue Periférico*. [S.l.: s.n.], 2013. Citado na página 17.
- NAOUM, F. A. N. P. C. Interpretação laboratorial do hemograma. 2007. Citado 3 vezes nas páginas 17, 20 e 21.
- NICK, M. *TensorFlow Machine Learning Cookbook : Over 60 Recipes to Build Intelligent Machine Learning Systems with the Power of Python, 2nd Edition*. Packt Publishing, 2018. Second edition. ISBN 9781789131680. Disponível em: <<http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1883903&lang=pt-br&site=ehost-live>>. Citado 2 vezes nas páginas 22 e 23.

OFFIR, B.; LEV, Y.; BEZALEL, R. Surface and deep learning processes in distance education: Synchronous versus asynchronous systems. *Computers and Education*, v. 51, n. 3, p. 1172–1183, 2008. ISSN 0360-1315. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0360131507001406>>. Citado na página 14.

RASCHKA, S. *Python Machine Learning*. Packt Publishing, 2015. (Community Experience Distilled). ISBN 9781783555130. Disponível em: <<http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1071004&lang=pt-br&site=ehost-live>>. Citado 2 vezes nas páginas 22 e 23.

RAVI, D. et al. Deep learning for health informatics. *IEEE Journal of Biomedical and Health Informatics*, v. 21, n. 1, p. 4–21, 2017. Citado na página 14.

SHENGGAN. *BCCD Dataset*. 2019. Citado 4 vezes nas páginas 18, 19, 20 e 32.

STREAMLIT. 2021. <<https://streamlit.io/>>. Acessado em: 2021-10-31. Citado na página 36.

SUGOMORI, Y. et al. *Deep Learning: Practical Neural Networks with Java*. Packt Publishing, 2017. ISBN 9781788470315. Disponível em: <<http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1532297&lang=pt-br&site=ehost-live>>. Citado na página 24.

TENSORFLOW. *Object Detection*. 2021. <https://github.com/tensorflow/models/tree/master/research/object_detection>. Acessado em: 2021-10-31. Citado na página 33.

TENSORFLOW. 2021. <<https://www.tensorflow.org/>>. Acessado em: 2021-06-04. Citado 2 vezes nas páginas 27 e 36.

VIVAS, W. L. P. *Manual Prático de Hematologia*. [S.l.: s.n.], 2017. Citado 4 vezes nas páginas 17, 18, 20 e 21.

ZHAO, R. et al. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, v. 115, p. 213–237, 2019. ISSN 0888-3270. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0888327018303108>>. Citado na página 14.

ZOCCA, V. et al. *Python Deep Learning*. Packt Publishing, 2017. ISBN 9781786464453. Disponível em: <<http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1513367&lang=pt-br&site=ehost-live>>. Citado na página 26.