



Universidade Paulista - UNIP

Ciência da Computação

TRABALHO DE CONCLUSÃO DE CURSO

ANÁLISE COMPARATIVA DE MODELOS DE REDES CONVOLUCIONAIS NA IDENTIFICAÇÃO DE CROMOSSOMOS EM IMAGENS DIGITAIS

Orientador: Paulo César Polastri

Márcio Rogério Spadari Júnior - N6110C0

Matheus Pereira da Silva - N610CH2

Nathan Lucas Santos Nicolau - F310FH5

Universidade Paulista - UNIP

São José do Rio Preto - 2023

MÁRCIO ROGÉRIO SPADARI JÚNIOR

MATHEUS PEREIRA DA SILVA

NATHAN LUCAS SANTOS NICOLAU

**Análise comparativa de modelos de redes convolucionais na
identificação de cromossomos em imagens digitais**

Trabalho de Conclusão de Curso
apresentado ao curso de Ciências da
Computação, tendo como objetivo a análise
comparativa sobre o desempenho de técnicas
utilizando redes neurais para detecção de
elementos em imagens.

UNIVERSIDADE PAULISTA - UNIP
SÃO JOSÉ DO RIO PRETO - 2023

MÁRCIO ROGÉRIO SPADARI JÚNIOR
MATHEUS PEREIRA DA SILVA
NATHAN LUCAS SANTOS NICOLAU

**Análise comparativa de modelos de redes convolucionais na
identificação de cromossomos em imagens digitais**

Trabalho de Conclusão de Curso
apresentado ao curso de Ciências da
Computação, para obtenção da graduação na
Universidade Paulista - UNIP orientado pelo
professor Paulo César Polastri.

São José do Rio Preto, __ de _____ de ____

BANCA EXAMINADORA

Primeiro examinador

Nome: _____

Instituição: _____

Segundo examinador

Nome: _____

Instituição: _____

Terceiro examinador

Nome: _____

Instituição: _____

AGRADECIMENTOS

Márcio:

Agradeço a todo o apoio e incentivo que tive para qual possa atingir meus objetivos, minha mãe Danubia, meu pai Márcio e namorada Ketelyn, por serem compreensivos e estar juntos com o foco no sucesso em todo esse processo, ao professor Paulo que nos auxiliou com dúvidas e esclarecimentos, ao professor Cícero, que nos auxiliou no quesito do aperfeiçoamento da documentação. E a todos que fizeram parte dessa caminhada.

Matheus:

Agradeço a toda minha família por estar presente neste momento da minha vida, onde me apoiaram e nunca deixaram eu desistir, agradeço também ao meu mentor Alexandre Marques que me deu todo apoio e suporte no meu desenvolvimento profissional e incentivou a fazer esse projeto. Ao professor Cícero e Paulo que apoiou e encaminhou o grupo para fazer este trabalho da melhor maneira possível. Agradeço também aos meus colegas de equipe deste projeto no qual trabalhamos juntos e em equipe para que fosse finalizado.

Nathan:

Agradeço primeiramente a Deus, pela capacidade que me deu de chegar até este momento, à minha esposa, que me compreendeu nas horas dedicadas nessa jornada, à minha filha, que sempre me motivou na busca de dar meu melhor, a meus pais, que acreditaram em meu sonho, e a todos os professores que pude conhecer nessa etapa da minha vida, que através das duas didáticas, sempre agregaram valores aos meus estudos e na minha formação como profissional.

RESUMO

O trabalho buscou definir bem alguns dos principais conceitos da área da Inteligência Artificial, na representação das Redes Neurais e Redes Convolucionais. A proposta central foi a de utilizar dois modelos para identificação de diferentes classes de cromossomos fornecidas, sendo um modelo desenvolvido com base na estrutura YOLOv8, outro no modelo VGG16, ambas utilizando pesos pré-estabelecidos do ImageNet. Um dos objetivos estabelecidos é analisar de forma comparativa, quais dos modelos, para a situação proposta, apresenta os melhores resultados nas metrificações de treinamento e de reconhecimento em novas imagens fornecidas. No qual apresentamos alguns pontos sobre porque optamos pelo uso do Yolo V8.

Palavras-chave: Redes Neurais, Redes Convolucionais, YOLOv8, VGG16, Metrificações de Treinamento, Reconhecimento de Imagens, Classificação de Cromossomos.

ABSTRACT

This work sought to define well some of the main concepts in the area of Artificial Intelligence, in the representation of Neural Networks and Convolutional Networks. The central proposal was to use two models to identify different classes of chromosomes provided, one model being developed based on the YOLOv8 structure, another in the VGG16 model, both using pre-established ImageNet weights. One of the established objectives is to analyze in a comparative way, which of the models, for the proposed situation, presents the best results in training and recognition metrics in new images provided. In which we present some points about why we chose to use the Yolo V8.

Keywords: Neural Networks, Convolutional Networks, Yolo V8, VGG16, Training Metrics, Image Recognition, Chromosome Classification.

LISTA DE FIGURAS

Figura 1: Diagrama do fluxo da metodologia abordada.....	17
Figura 2: Representação em cores de um cromossomo.....	23
Figura 3: Representação da estrutura de um neurônio do cérebro humano.....	24
Figura 4: Representação da estrutura do neurônio artificial proposto.....	24
Figura 5: Representações matemáticas do Neurônio Artificial.....	25
Figura 6a e 6b: Representação da Função Degrau.....	28
Figura 7a e 7b: Representação da Função Sigmóide em formato de s.....	28
Figura 8: Representação de uma Rede de camada única.....	31
Figura 9: Representação de uma estrutura de Rede recorrente.....	31
Figura 10: Representação de uma estrutura de Rede recorrente.....	32
Figura 11: Representação simplificada de uma estrutura de rede convolucional.....	35
Figura 12: Imagem 103064 do dataset utilizado.....	38
Figura 13: Representação das classes de cromossomos utilizados.....	39
Figura 14: Organograma que apresenta fluxo de processos das imagens.....	41
Figura 15: Diagrama sobre o fluxo do modelo Yolo.....	44
Figura 16: Imagem após processo de delimitação supervisionada.....	45
Figura 17: Resultado de detecção em uma nova imagem.....	53

LISTA DE GRÁFICOS

Gráfico 1: Comparativo YOLOv8 com as versões anteriores.....	36
Gráfico 2: Resultados de treinamento do YOLOv8 na máquina 1.....	48
Gráfico 3: Resultados de treinamento do YOLOv8 na máquina 2.....	50
Gráfico 4: Resultados da métrica F1 que combina precisão e revocação.....	52
Gráfico 5: Resultados da métrica da curva de precisão e confiança.....	52
Gráfico 6: Porcentagem de acurácia no treinamento e validação das imagens.....	54
Gráfico 7: Porcentagem de perda das imagens de treinamento e validação.....	54

LISTA DE TABELAS

Tabela 1: Representação dos dados da primeira época de treinamento.....	43
Tabela 2: Resultados obtidos do YOLOv8 treinado em ambiente com GPU.....	47
Tabela 3: Resultados obtidos do YOLOv8 treinado em ambiente com CPU.....	49
Tabela 4: Ambiente de execução em GPU - Máquina 1.....	51
Tabela 5: Ambiente de execução em CPU - Máquina 2.....	51
Tabela 6 Representação dos resultados obtidos por época.....	55

LISTA DE SIGLAS

AM	Aprendizado de Máquina
AP	Algoritmo de Aprendizado
CLI	Campo Local Induzido
IA	Inteligência Artificial
RC	Rede Conexionista
RNA	Rede Neural Artificial
RN	Rede Neural
RNC	Rede Neural Convolucional
ROI	Região de Interesse
IoU	Interseção sobre União

SUMÁRIO

1 INTRODUÇÃO	12
1.1 Objetivos Gerais	12
1.2 Objetivos Específicos	13
1.3 Análises dos Resultados e Comparações	14
1.4 Justificativas	15
1.5 Metodologia	17
2 REFERENCIAL TEÓRICO	21
2.1 Utilização de Redes Neurais Artificiais	21
2.2 Citogenética e os cromossomos	22
2.3 Modelo de neurônio artificial	23
2.4 Redes neurais artificiais	26
2.4.1 Algoritmo backpropagation	26
2.4.2 Funções de ativação	27
2.4.3 Modelo Perceptron	29
2.4.4 Topologia	30
2.5 Redes neurais convolucionais	32
2.5.1 Modelos de redes convolucionais utilizadas	35
2.6 Método Yolo (You Only Look Once)	36
2.7 Método VGG16	36
2.8 Dataset	37
2.9 Classes dos cromossomos identificados	39
3 DESENVOLVIMENTO	40
3.1 Ambientes de Execução	40
3.2 Processo de ajuste do dataset para a solução VGG16	41
3.3 Processo de treinamento do modelo VGG16	42
3.4 Processo de ajuste do dataset para o modelo YOLOv8	44
3.5 Processo de classificação das imagens para o modelo YOLOv8	45
3.6 Processo de treinamento do modelo YOLOv8	46
3.6.1 Treinamento feito na Máquina 1:	46
3.6.2 Treinamento feito na Máquina 2:	48
4 ANÁLISE DOS RESULTADOS OBTIDOS	51
4.1 Resultados obtidos no modelo YOLOv8	51
4.2 Resultados obtidos com o modelo VGG16	54
5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	57
6 BIBLIOGRAFIA	59

1 INTRODUÇÃO

Historicamente, sabe-se que os computadores foram criados com o intuito de quebrar os meios de comunicação militar, mas, com o passar das décadas, suas aplicações atingiram escaláveis evoluções, nas mais diversas áreas de pesquisa.

A crescente evolução das máquinas, do estudo aplicado a soluções mais complexas, e do exponencial uso dos recursos computacionais não só para cálculos de lógica matemática, mas para tarefas envolvidas em automatizações e visão computacional, promoveram o nascimento da ramo da Inteligência Artificial (IA), tornando-a com o passar do tempo, uma das mais áreas com maior evolução de pesquisa em área acadêmica e em uso comercial. Segundo palavras de Satya Nadella, atual CEO da Microsoft, durante a Microsoft Envision, 2016, graças às formas de automação dos sistemas inteligentes e do próprio uso de inteligência artificial, as empresas têm se transformado numa velocidade incrível, alterando desde os modelos de negócios adotados, às formas de trabalho e à convivência humana.

Grande parte desse fator se deve ao aprendizado de máquina(AM), que de certa forma, é a porção de estudo dos sistemas inteligentes, incubida de proporcionar formas de aprendizado aos computadores, aplicando diferentes formas de raciocínio, deduções e tomadas de decisão. Dentre suas utilizações mais comuns, destacam-se as formas de automação de processos com utilização de redes neurais, que não apenas aumentam a eficiência operacional, mas também reduzem os riscos de falhas humanas em diversos processos, podendo ser aplicados desde tarefas de baixa complexidade, como a criação de planilhas e relatórios, a processos do campo da visão computacional, como análises de imagens na área médica.

1.1 Objetivos Gerais

O assunto abordado parte da aplicação de dois modelos de Redes Convolucionais distintos, sendo submetidos a treinamentos supervisionados com um dataset de cromossomos previamente selecionado. Como proposta central, foi

realizada uma análise comparativa, averiguando os resultados obtidos por ambas as estruturas, na identificação de 24 classes distintas de cromossomos, e por fim comparamos os resultados obtidos.

1.2 Objetivos Específicos

Com base no objetivo central, torna-se crucial avaliar o desempenho das arquiteturas empregadas no processo de identificação dos cromossomos, levando em consideração a velocidade de processamento, a eficiência na precisão do reconhecimento, a complexidade da configuração, a facilidade de modulação dos parâmetros e a inserção de dados, bem como a dificuldade associada à metrificar as saídas.

Para atingir esse objetivo específico, dimensionamos os objetivos em tópicos principais, tais como:

Análise de Desempenho das Arquiteturas:

- Avaliação da velocidade de processamento das arquiteturas consideradas.
- Análise da eficiência na precisão do reconhecimento de cromossomos para cada arquitetura.
- Comparação dos resultados obtidos em estudos anteriores e suas implicações para a aplicação proposta.

Configuração e Otimização das Redes Neurais:

- Compreensão detalhada de cada etapa no desenvolvimento de redes neurais para o reconhecimento de cromossomos.
- Estratégias de configuração para atingir o melhor desempenho, considerando a natureza específica do problema.

Modularidade de Parâmetros e Inserção de Dados:

- Avaliação da facilidade de modularidade dos parâmetros das arquiteturas consideradas.

- Investigação sobre a eficácia da inserção de dados e a flexibilidade oferecida pelas diferentes arquiteturas.

Desafios e Soluções na Mensuração das Saídas:

- Identificação dos desafios associados à mensuração das saídas das arquiteturas.
- Proposta de soluções para superar obstáculos na avaliação e interpretação dos resultados obtidos.

Ao abordar esses tópicos de maneira abrangente, foi possível realizar uma análise completa do desempenho das arquiteturas de redes neurais e convolucionais no contexto específico da identificação de cromossomos, atendendo assim ao objetivo estabelecido.

1.3 Análises dos Resultados e Comparações

Uma vez selecionado o dataset de treinamento que contém as imagens, foram feitos alguns processos, relatados posteriormente, para adaptação do uso das imagens nas etapas de treinamento e teste das redes.

- Realizar testes de treinamento e validação, avaliando o desempenho das redes neurais convolucionais em termos de velocidade de processamento e eficiência, no que diz respeito principalmente à precisão na identificação dos cromossomos.
- Analisar os resultados obtidos, comparando não somente as diferentes arquiteturas, mas também a diferença entre os ambientes de execução dos modelos e das máquinas utilizadas para tais processos, identificando os melhores resultados apresentados.

- Realizar diferentes sequências de testes, com ajustes de parâmetros de treinamento, como dimensão e quantidade de imagens fornecidas por vez ao treinamento da rede.
- Investigar os gastos computacionais e o tempo de processamento envolvidos nas arquitetura testadas, considerando a viabilidade prática de implementação.
- Discutir as aplicações reais de identificação automatizada de cromossomos na área da saúde, levando em conta os resultados obtidos e as limitações encontradas.
- Apresentar conclusões e recomendações com base na análise dos resultados, indicando possíveis melhorias e direções para pesquisas futuras na área.

Ao cumprir esses objetivos específicos, espera-se contribuir para o avanço da contagem automatizada de cromossomos, fornecendo insights sobre o desempenho de diferentes arquiteturas de redes neurais convolucionais (RNCs) e sua viabilidade para aplicação prática na área da saúde.

1.4 Justificativas

Segundo ANDREW (2017), estudioso e fundador do projeto Google Brain a revista Fortune, podemos considerar a Inteligência Artificial como a nova eletricidade, no sentido de que, assim com todas as formas de produção e convivência humana foram transformadas por ela no início do século 20, usufruiríamos um mesmo impacto de mudança em toda a indústria, só que numa escala de tempo de 9 vezes menor.

Com o passar dos anos, e com a atual disponibilização de ferramentas poderosas baseadas em algoritmos de aprendizagem profunda, como o Chat GPT e Google Bard, torna-se visível a eficiência das máquinas com algoritmos de

aprendizado, na distribuição de informações, notícias, auxílios de estudo, no entretenimento digital e em tantas outras áreas de conhecimento e pesquisa acadêmica.

Conforme definido por MOREIRA (2002) pelas características de construção de estrutura das RNAs proporcionarem a elas excelentes resultados de atribuição classificatória, grande parte de suas aplicações, dentro da área da Inteligência Artificial, está voltada à alguma forma de reconhecimento de padrões, sendo em aplicações de área médica, de comunicação e tradução, ou ainda em sistemas autônomos. Dessa forma, é quase certo especular que os mais diversos processos de identificação de alguma forma mínima de padrão, terão cada vez mais alguma relação a algum tipo de sistema de aprendizado.

Dentre as áreas que podem ser destacados os crescimentos de algoritmos de IA, podemos abstrair o ramo da biologia, mais especificamente, no campo dos estudos de estruturas do DNA humano, onde se enquadram os cromossomos.

Atualmente, as formas utilizadas para tal detecção, em grande maioria, são realizadas por profissionais da área de estudo da genética humana, conhecida como citogenética, que se refere a todo e qualquer estudo relativo ao cromossomo isolado ou em conjunto (cariótipo).

Em NICOLAU e CHAVES (2013), temos a informação de que um dos métodos de identificação manual é a hibridização *in situ* por fluorescência, que envolve a preparação de lâminas, o isolamento e a marcação da sequência de DNA que se deseja localizar *in situ* e a sua hibridização nos cromossomos.

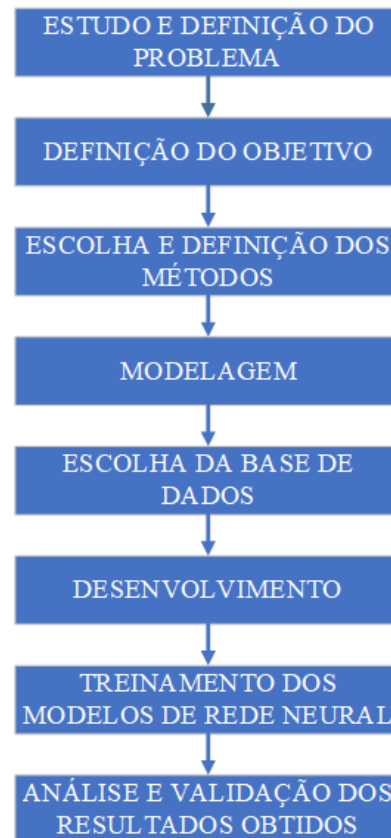
Esse estudo é feito por agentes humanos, e justamente por isso, algumas variáveis acabam por influenciar essa tarefa, como possíveis distrações recorrentes, perda de foco, cansaço e ainda outros fatores externos, como a avaliação individual das imagens a serem utilizadas por cada profissional.

É justamente pela inexistência desses fatores num processo computacional, que o uso de algoritmos de redes neurais com técnicas de aprendizado supervisionado se mostram muito eficientes.

1.5 Metodologia

Tomando como base as metodologias discriminadas em CIMIRRO (2022) foram definidas as seguintes estruturas base para modelagem e construção dos algoritmos propostos.

Figura 1: Diagrama do fluxo da metodologia abordada



Fonte: Autor (2023)

Para a primeira etapa das atividades realizadas, envolvendo o estudo e definição do problema, foi necessário uma análise detalhada do histórico de detecção do objeto, bem como um estudo detalhado dos cromossomos. Esta fase envolve uma extensa revisão da literatura científica relevante para compreender os desafios enfrentados por outros investigadores, como abordam o problema e para identificar as chamadas “lacunas de conhecimento”, essa fase inicial, proporcionou uma base sólida para as etapas subsequentes da metodologia.

A segunda etapa da metodologia envolve a definição precisa do problema em foco. Com base na análise detalhada realizada na etapa anterior, que identificou os detalhes e desafios associados à detecção cromossômica em imagens digitais, esta

etapa visa delinear claramente os parâmetros e características básicas do problema.

Durante o processo de definição do problema foi estabelecido um escopo específico, incluindo critérios de inclusão e exclusão das imagens a serem consideradas. Além disso, as principais variáveis que afetam a detecção cromossômica foram identificadas para fornecer uma base sólida para o desenvolvimento subsequente da rede neural convolucional (CNN).

A terceira atividade, no estudo de métodos analíticos, conduziu uma análise abrangente dos métodos, ferramentas e algoritmos existentes de identificação de cromossomos. Durante este processo, foram examinadas sua eficiência e características específicas com o objetivo de adaptar e selecionar o método mais adequado para atingir os objetivos propostos neste trabalho.

Após a conclusão da pesquisa experimental para obtenção da fundamentação teórica, inicia-se o processo de seleção dos trabalhos relevantes. Neste contexto, este estudo tem como objetivo identificar literaturas científicas empregando técnicas de detecção cromossômica em imagens digitais.

Após isso foram dados os seguintes passos:

1. Levantamento bibliográfico com uma pesquisa experimental para fundamentação teórica com foco na identificação de métodos existentes para detecção de cromossomos em imagens digitais utilizando inteligência artificial.
2. Desenvolvimento de um processo de seleção de trabalhos correlatos. Utilização de técnicas de busca em documentos científicos, priorizando o emprego de inteligência artificial na detecção de cromossomos.
3. Seleção da base de busca utilizando o Google Scholar devido à sua abrangência em resultados de congressos e eventos relevantes, documentos e materiais sugeridos e fornecidos pelos professores da universidade sobre rede neural convolucional.

4. Citação e utilização dos trabalhos escolhidos para embasar, justificar e responder às questões de pesquisa.

A terceira etapa do método envolve a seleção de uma abordagem para resolver o problema definido. Neste projeto optamos por utilizar a linguagem de programação Python devido à sua versatilidade e ampla gama de bibliotecas disponíveis para visão computacional e aprendizado de máquina.

Para detecção de objetos, dois modelos diferentes foram implementados: YOLOv8 e VGG16. YOLOv8 (You Only Look Once, versão 8) é conhecido por sua eficiência e velocidade, tornando-o uma escolha adequada para tarefas de detecção em tempo real.

A implementação desses modelos foi feita de forma adequada, usando estruturas e bibliotecas estabelecidas na comunidade de aprendizado de máquina, como PyTorch e TensorFlow. Este processo permitirá uma análise comparativa do desempenho entre os dois modelos selecionados.

O dataset escolhido para esse projeto conta com uma gama alta de imagens cromossômicas que, pela sua relevância e representatividade, motivou e foi um dos fatores principais que pesaram, considerando a qualidade e diversidade das imagens como fatores essenciais para o treinamento e avaliação dos modelos.

A integração das imagens da base de dados com os modelos YOLOv8 e VGG16 foi cuidadosamente planejada, considerando a representatividade da base em relação ao problema definido.

Durante a sexta etapa, que se trata do desenvolvimento prático dos modelos, YOLOv8 e VGG16, foram treinadas utilizando o dataset selecionado. O processo de treinamento envolve alimentar imagens na rede, permitindo que o modelo aprenda a identificar e localizar cromossomos.

Para o treinamento e validação dos algoritmos de Inteligência Artificial, foi escolhido executar localmente na IDE Pycharm no qual foi feita a análise utilizando hardwares e placas gráficas diferentes para cada modelo, uma diferença entre os trabalhos mencionados neste projeto, no qual utilizam o Google Colab que fornece o processamento do algoritmo em GPU (Graphical Processing Unit) de alto desempenho sendo executado em nuvem.

2 REFERENCIAL TEÓRICO

Para poder estabelecer corretas definições literárias, e aplicar boas metodologias de desenvolvimento, bem como abordagens de apuração de resultados de forma eficiente, foram utilizadas como base para esse trabalho de conclusão de curso outras produções acadêmicas, como artigos, monografias, e materiais de estudo acadêmico, sendo os principais ANDRADE (2019), CIMIRRO (2022), MOREIRA (2002) e SANTOS, JUNIOR, WADA, TENÓRIO, BARBOSA E MARQUES (2019) Também foi utilizada como fonte principal de estudo de Redes Neurais o livro: Redes Neurais: Princípios e Práticas HAYKIN (2001).

2.1 Utilização de Redes Neurais Artificiais

As RNAs, ou Redes Conexionistas, pertencem à área de estudo da Inteligência Artificial, e representam grande parte de sua pesquisa. Sua origem se relaciona diretamente ao estudo biológico do funcionamento do cérebro humano e seus neurônios, de forma a replicar sua estrutura.

Levando em conta a eficiência e velocidade em processos de reconhecimento já mencionados anteriormente, HAYKIN (2001) baseou suas justificativas nas características de construção presentes nas redes neurais, propondo a explicação de que uma RNA poderia ser basicamente definida como a união de pequenas unidades de processamento, cada qual operando paralelamente, mas que são unidas a fim de constituir um grande processador, com o objetivo de armazenar e repassar informações.

Conforme dito anteriormente, o uso das RNAs é amplamente destinado a tarefas de reconhecimento, dentre essas, uma de grande notoriedade seria a de reconhecimento de imagens, ou ainda, objetos que fazem parte de sua composição. Isso se dá pela utilização de técnicas de processamento de imagens digitais. Foi a parte desse contexto, que se definiu o tema desse Trabalho de Conclusão de Curso.

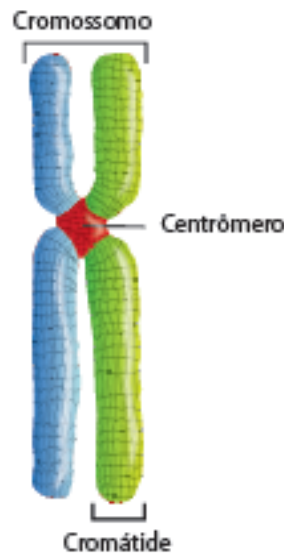
2.2 Citogenética e os cromossomos

Os cromossomos, identificados como estruturas que compõem o DNA humano, são os responsáveis por carregar o material genético, fornecido pelos progenitores, HOSPITAL EINSTEIN (2022). Cada um é uma entidade compacta, composta por uma longa e linear molécula de DNA associada a proteínas especializadas. Essas estruturas notáveis estão alojadas no núcleo das células eucarióticas, que são as células com núcleos definidos. A quantidade de cromossomos varia de uma espécie para outra. Na espécie humana, por exemplo, eles são organizados por:

- **Células Somáticas:** que são as células não reprodutivas do nosso corpo, carrega 46 pares de cromossomos homólogos. A combinação desses cromossomos ajuda a garantir a estabilidade genética nas células do corpo humano.
- **Células Germinativas ou Gametas:** As células reprodutivas, como os espermatozóides e os óvulos, possuem apenas 23 cromossomos, o que é metade do número presente nas células somáticas.

Um ponto importante para se destacar em relação a esses, é em qual processo de divisão celular eles se tornam mais facilmente perceptíveis. Segundo MUNIZ (2001), dentre todas as etapas, é na mitose, que as estruturas dos cromossomos se tornam mais identificáveis, sendo ela o processo de divisão celular que leva à formação das células diplóides, isto é, com um conjunto cromossômico oriundo do pai e outro da mãe. A citogenética, pertencente à área da biologia, é a ciência que estuda os cromossomos. Nas palavras de MUNIZ (2011) ela desempenha um papel muito importante no auxílio de diagnósticos de doenças cromossomopatias, ou seja, as que atingem diretamente os cromossomos, como algumas síndromes relacionadas com as trissomias dos cromossomos 21,18,13,5,X e Y.

Figura 2: Representação em cores de um cromossomo.



Fonte: Adaptado de MUNIZ (2011)

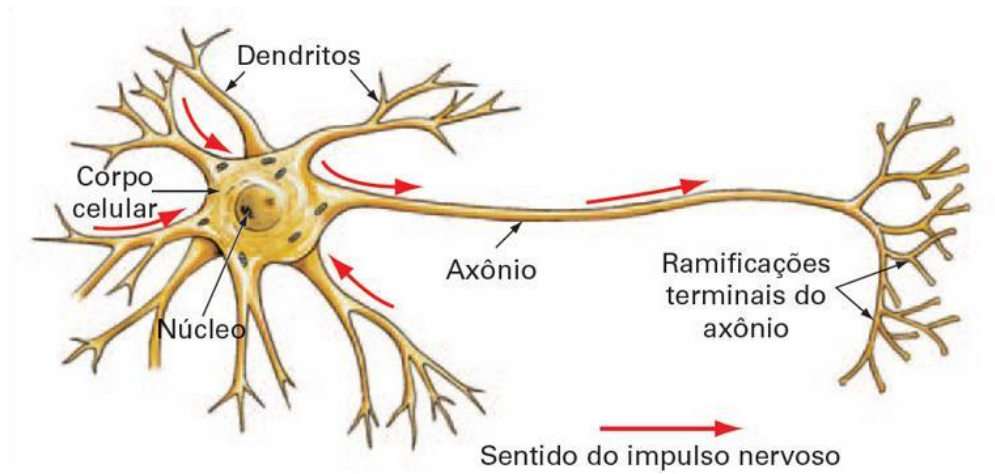
Sabendo de antemão do benefício a área da saúde do estudo dos cromossomos, SANTOS e MARANA (2019) pontuaram que o uso de algoritmos de processamento de segmentação de imagens digitais com o objetivo de classificação, unidos ao uso de Redes Neurais, têm apresentado resultados promissores.

2.3 Modelo de neurônio artificial

Dentre os modelos desenvolvidos inicialmente para a construção de um neurônio artificial, o que obteve maior sucesso foi o proposto por MCCULLOCH, PITTS (1943), chamado de *Perceptron*, seguindo sua construção com algoritmos matemáticos.

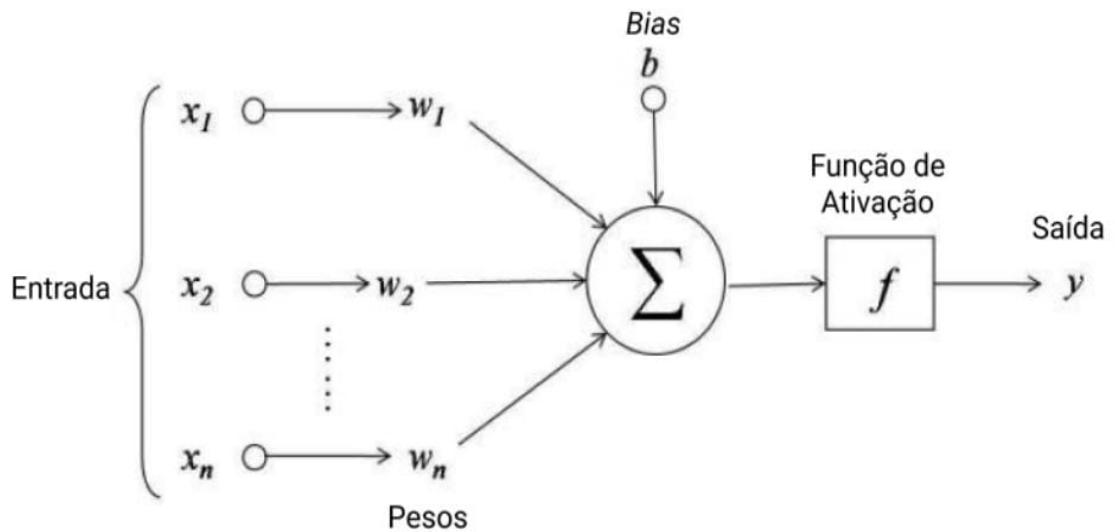
Seu funcionamento baseado em cálculos matemáticos, posteriormente lhe atribuiria a capacidade de computar operações de lógica computacional, como “or” e “and”. De forma simples, pode-se observar pequenas semelhanças entre as duas concepções de neurônios, ilustradas nas Figuras 3 e 4:

Figura 3: Representação da estrutura de um neurônio do cérebro humano



Fonte: Livro digital Deep Learning Book, Capítulo 4 – O Neurônio, Biológico e Matemático.

Figura 4: Representação da estrutura do neurônio artificial proposto originalmente por McCulloch e Pitts (1943)



Fonte: Adaptada de HAYKIN (2001)

Segundo a definição dada por HAYKIN (2001) nessa estrutura temos:

- As *sinapses* ou *elos de conexão*, representados cada um com seus respectivos *pesos* próprios.

- O sinal X_m na entrada da sinapse m conectado ao neurônio k é multiplicado por seu respectivo peso sináptico W_{kj} . Nesse índice, temos k representando o neurônio e j atribuído ao terminal de entrada da sinapse, no qual o peso se refere.
- Um *somador* Σ para somar os sinais de entrada, sendo ponderados pelas respectivas sinapses dos neurônio. Vale destacar que as operações descritas aqui constituem um *combinador linear* tendo U_k representando sua saída.
- Uma *função de ativação* $\varphi(\cdot)$ para restringir a amplitude da saída de um neurônio k . Ela também pode ser referida como uma função restritiva, já que serve também para limitar a amplitude do sinal de saída dentro de um valor finito
- *bias* b_k tem o efeito de aumentar ou diminuir os valores das entradas da função de ativação, dependendo se tem valor positivo ou negativo.

Na sua construção matemática, HAYKIN (2001) transcreveu o seguinte algoritmo:

Figura 5: Representações matemáticas do Neurônio Artificial da Figura 4

$$u_k = \sum_{j=1}^m w_{kj} \cdot x_j$$

$$Y_k = \varphi(U_k)$$

Fonte: Adaptado de HAYKIN (2001)

Onde, seguindo como base as mesmas identificações atribuídas anteriormente, temos:

- X_m representando os sinais de entrada;
- W_{km} são os pesos sinápticos do neurônio k ;
- U_k é identificado como a saída do *combinador linear* devido aos sinais de entrada;

- b_k é o *bias*;
- $\phi(\cdot)$ é a função de ativação
- Y_k é a saída do neurônio;

2.4 Redes neurais artificiais

Segundo HAYKIN (2001), uma RNA pode ser entendida como um processador maciçamente paralelo e distribuído, constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso.

Sobre seu surgimento ele trouxe o contexto dos estudos existentes no ramo da biologia sobre o cérebro humano, no que diz respeito a sua estrutura física e seu funcionamento. Sua definição, apresentou o cérebro e suas características de processamento de informações, sendo similar a um grande processador, que, conforme processadas novas operações e adquiridos novos conhecimentos, aprimora sua forma de raciocínio. Conceitualmente, essa estrutura formada por neurônios, utilizando comunicação por sinapses (impulsos elétricos), se rearranja a modo de proporcionar o mais eficiente funcionamento de todos os variados tipos de tarefas. Entretanto, o que se tornou mais aparente em eficiência, foram os resultados dos processamentos de reconhecimentos cognitivos, como os reconhecimentos de padrões, rotinas, identificação de rostos familiares ou objetos de convivência cotidiana.

Mediante a essa definição, o estudo das RNAs, dentro da área da IA, tomou como objetivo replicar de forma lógica essas estruturas, a fim de reproduzir seu funcionamento, visando uma eficiência similar nos resultados obtidos.

2.4.1 Algoritmo backpropagation

Segundo HAYKIN (2001) as maiores similaridades de uma RNA, quando comparada a estrutura do cérebro humano, está no processo de aprendizagem. Isso se deve ao fato de que em ambos os lados, o conhecimento é adquirido através de

algum *processo de aprendizagem*, e de que as forças de conexão (pesos) entre os neurônios constituintes, são os responsáveis por armazená-lo.

Pensando no lado das redes neurais, temos o chamado *algoritmo de aprendizagem*, cuja função é modificar os pesos sinápticos, ou seja, ajustar os valores, que percorrem os neurônios nas camadas da rede de forma ordenada, visando alcançar o objetivo de promover um resultado mais próximo ao esperado na camada de saída.

O algoritmo *Backpropagation*, ou ainda *error backpropagation*, é um dos algoritmos de aprendizagem mais aplicados na construção de RNAs mais simples, trabalhando com ajustes nos “erros” encontrados. Basicamente, sua funcionalidade pode ser explicada com duas etapas:

- Primeiro, é dado um **passo à frente**, no qual a entrada é processada através das camadas ocultas até a camada de saída, estimando desta forma um meio de uma previsão do resultado.
- Posteriormente, um **passo atrás** onde ocorre uma retropropagação do erro, no qual é calcular a diferença entre a saída estimada e o valor esperado, esta diferença é utilizada para o ajuste dos pesos sinápticos, desde a camada de saída até a entrada, de forma a minimizar o erro no processo de iteração e treinamento.

Essas duas etapas se repetem iterativamente durante o treinamento de uma RNA, com o objetivo de otimizar os pesos sinápticos de modo que a rede neural seja capaz de fazer previsões mais precisas e se aproxime cada vez mais dos resultados desejados.

2.4.2 Funções de ativação

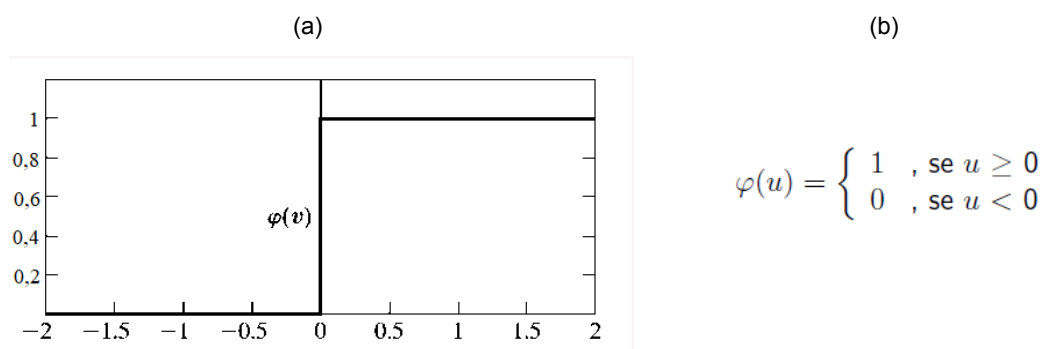
Seguindo as identificações atribuídas no modelo de neurônio artificial ilustrado na Figura 5, são apresentados alguns tipos de funções de ativação que podem ser aplicados na construção de uma RNA. Quando falamos dessas funções, estamos tratando do algoritmos matemáticos que são aplicados aos neurônios das camadas da rede, segundo HAYKIN (2001) pontua-se que, a função de ativação atua recebendo o valor provido pela junção aditiva - que agora recebe o nome de

campo local induzido (CLI) - somado já ao valor do bias b_k , atribuindo 1, caso o valor fornecido pelo CLI for positivo, ou 0 caso contrário. Essa função de ativação é a aplicada inicialmente no modelo de neurônio de MCCULLOCH e PITTS (1943), recebendo o nome de *Função Degrau*, ou ainda *Função de Limiar* (Figura 4).

Existem ainda outras duas funções nomeadas por HAYKIN (2001) sendo a *Função Linear por Partes*, que pode ser vista como uma forma de aproximação de um amplificador não linear, e a *Função sigmóide*, que dentre as três apresentadas, é certamente a mais utilizada quando se desenvolve uma RNA. Seu comportamento segue uma ordem crescente, exibindo um balanceamento entre os comportamentos dito lineares e não-lineares. Uma de suas representações matemáticas pode ser definida na *Função Logística*.

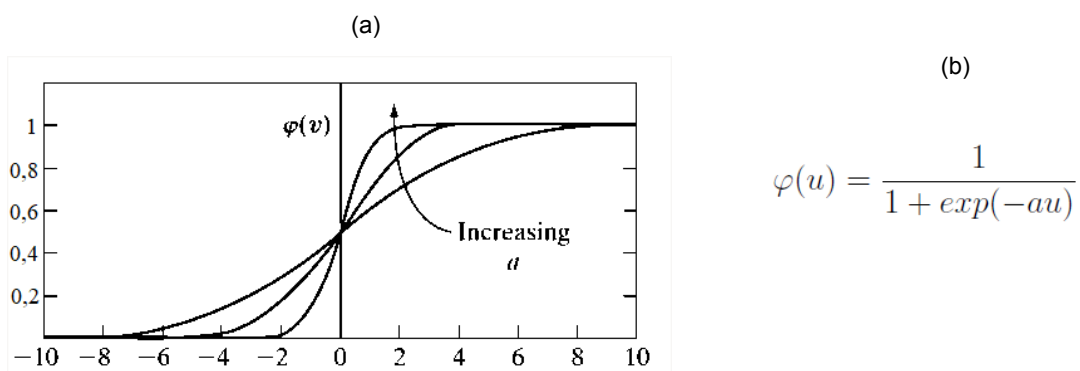
Ilustrativamente, ambas as Funções podem ser vistas em gráficos e algoritmos matemáticos:

Figura 6a e 6b: Representação da Função Degrau ;



Fonte: Adaptadas de HAYKIN (2001).

Figura 7a e 7b: Representação da Função Sigmóide em formato de s;



Fonte: Adaptadas de HAYKIN (2001)

A utilização da Função Sigmoidal se justifica pelo fato de apresentar um intervalo de valores capaz de proporcionar uma maior diferenciabilidade na RNA. Quando comparadas diretamente, observa-se que a *Função Degrau* assume unicamente os valores de 0 ou 1, enquanto que a *Função Sigmóide* valores entre 0 e 1. Essas características podem ser observadas nos respectivos gráficos da Figura 6 e Figura 7, onde a representação da crescente sigmoidal apresenta um formato de “S”.

2.4.3 Modelo Perceptron

Após a conceituação de sucesso da estrutura de um neurônio artificial, ROSENBLATT (1958) apresentou o modelo de *Rede Perceptron*, como o primeiro a ser aplicado com a aprendizagem supervisionada.

Também nomeado de *Perceptron de Camada Única*, Rosenblatt definiu tal modelo, utilizando como base o neurônio artificial de MCCULLOCH e PITTS (1943), como a forma mais simples de se construir uma RNA, usada para a classificação de padrões dito linearmente separáveis, ou seja, com a capacidade de reconhecimento aplicado diretamente associado à identificação de somente duas únicas classes distintas. O Perceptron, na ideia proposta por Rosenblatt, era uma representação inicial da capacidade das RNAs de aprender e tomar decisões com base em entradas de dados. No entanto, ele tinha limitações notáveis, pois só era capaz de lidar com problemas linearmente separáveis, o que significa que não era adequado para tarefas mais complexas que envolviam a separação de classes não lineares. Apesar dessas limitações, o modelo de *Perceptron de Camada Única* desempenhou um papel crucial no desenvolvimento da teoria das redes neurais artificiais e na compreensão inicial de como as máquinas podem aprender com dados. Posteriormente, um passo de sua evolução foi o *Multi Layer Perceptron*, aplicando seu conceito inicial com mais camadas de neurônios, com a principal diferença na utilização de camadas ocultas (Figura 6) .

Com o passar do tempo e a evolução de estudos, houve o surgimento de redes neurais mais complexas e algoritmos de treinamento mais avançados, tornando as RNAs capazes de abordar uma ampla gama de tarefas de aprendizado,

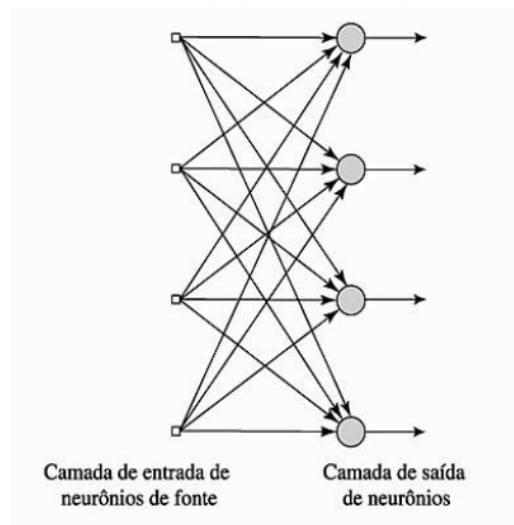
incluindo problemas não linearmente separáveis, culminando em avanços significativos na inteligência artificial.

2.4.4 Topologia

Segundo HAYKIN (2001) a topologia, ou ainda, as diferentes arquiteturas que podem ser aplicadas no desenvolvimento de uma RNA, estão intimamente ligadas ao algoritmo de aprendizagem que será aplicado a ela. Casualmente, são três tipos existentes:

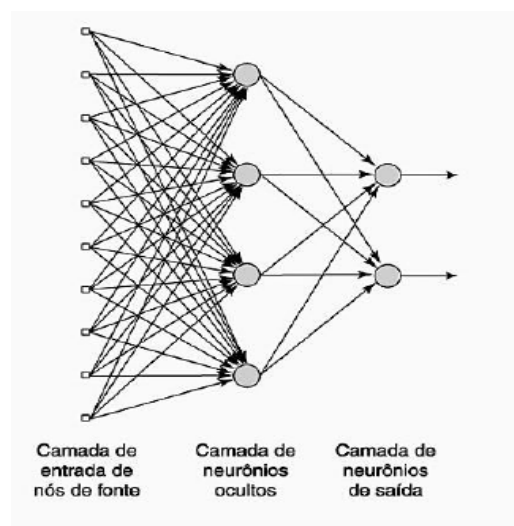
- **Redes Alimentadas Adiante (Feed-forward networks):** É comumente chamada de Rede acíclica, por na sua representação de grafos orientados não possuir nenhum ciclo. Nessa arquitetura, a rede é organizada de forma que os neurônios de sua composição, são distribuídos em camadas, tendo como fator fixo que o fluxo de informações sempre segue o sentido *adiante* (daí seu nome), ou seja, da camada de entrada até a camada de saída. HAYKIN (2001) destaca que, nesse tipo de estrutura, existem as *Redes de camada Única* e as *Redes de múltiplas camadas*, tendo ciência de que a diferença entre elas está unicamente na quantidade de camadas existentes, sendo que a primeira, ilustrada no exemplo da Figura 9, possui somente a camada de entrada e a camada de saída. Já no caso da de múltiplas camadas, ilustrada em exemplo na Figura 10, observa-se o uso de camadas ocultas. Independente desses fatores, o princípio de funcionamento adiante é o mesmo.

Figura 8: Representação de uma Rede de camada única.



Fonte: Adaptada de HAYKIN (2001)

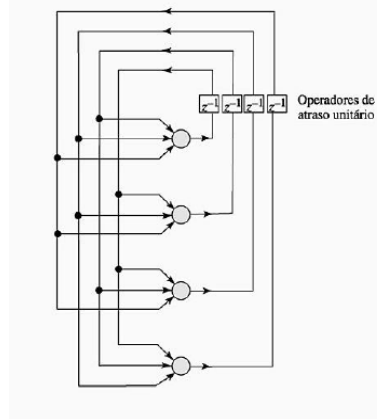
Figura 9: Representação de Rede de Múltiplas camadas.



Fonte: Adaptada de HAYKIN (2001)

- **Redes recorrentes (Feed-backward networks):** HAYKIN (2001) destaca que a diferença dessa arquitetura, quando comparada a uma rede múltipla de alimentação adiante, está no fato de que possui minimamente, um laço de realimentação. Basicamente, nessa estrutura, que não possui camadas ocultas, um determinado sinal de saída do neurônio n_1 serve como entrada (alimentação) para ele próprio ou ainda para outros neurônios de camadas anteriores. Um exemplo pode ser visto na Figura 10:

Figura 10: Representação de uma estrutura de Rede recorrente



Fonte: Adaptada de HAYKIN (2001)

- **Redes Competitivas:** Conforme apresentado por GONÇALVES (2008), essa variação de topologia de uma rede neural, é constituída de duas camadas de neurônios, sendo a *camada entrada* (ou ainda nós fontes) e a camada de saída, mais normalmente identificada como camada *grade*. As redes construídas com base nesse modelo costumam utilizar algoritmos de aprendizagem competitivo, ou seja, os neurônios de sua composição são “forçados” a competir uns com os outros, utilizando de base o nível de similaridade do padrão identificado na entrada e a determinada grade de neurônios. Dessa forma, somente o neurônio “vencedor” será ativado, para por fim, prosseguir com o fornecimento da sequência de dados.

2.5 Redes neurais convolucionais

Como anteriormente pontuado, a utilização de Redes Neurais em processos de reconhecimento cognitivos proporcionam resultados de notável eficiência de classificação. No entanto, segundo JURASZEK (2014), no que diz respeito a classificação de imagens, ou ainda de objetos, quaisquer que sejam de sua composição, temos alguns pontos-chaves a serem considerados. Um deles, seria a decisão de qual forma se tem a melhor representação de seus dados, ou seja, de suas características, sendo considerados de forma irrelevante detalhes como iluminação, orientação, pose, entre outros. Ele ainda acrescentou que, como proposta ideal de um sistema inteligente de classificação de imagens digitais, seria

necessário que ele fosse capaz de cumprir determinado processo de forma automática, em cima dos dados brutos, que nesse caso, seriam justamente as próprias imagens. Complementarmente, seguindo como base os estudos feitos por HUBEL e WIESEL (1968), ele apresentou a hipótese de uma representação hierárquica de uma determinada imagem digital, quando aplicada a processos de reconhecimento, partindo dos valores identificados individualmente de cada pixel, posteriormente de contornos, objetos, e por fim, cenas completas da imagem.

Esta estrutura sugere que o mecanismo de reconhecimento necessita de diversos estágios de treinamento empilhados uns nos outros, um para cada nível de hierarquia JURASZEK (2014).

Foi partindo desse contexto que surgiram as Redes Neurais Convolucionais (RNC) - ou ainda comumente chamadas de *Convnets* - utilizando como base as RNAs, mas com o foco de aplicação a formas de aprendizagem profunda, que necessariamente não precisassem do uso de dados previamente rotulados para treinamento. A principal diferença de uma RNC, quando comparada a uma RN convencional, está no fato de que sua estrutura é formada por densas camadas de neurônios, tendo atribuições diferentes de processamento para cada uma delas.

Conceitualmente são três os tipos de camadas que uma RNC possui, sendo a primeira uma camada de convolução, posteriormente uma camada de agrupamento e por fim uma camada totalmente conectada.

Dentre essas, é unanimemente definido que a camada de convolução torna-se a mais importante, pois sua função trata de aplicar máscaras (filtros) que percorrem toda a imagem fornecida através do uso da vizinhança de pixels, com objetivo gerar matrizes de dados, construídas com os valores correspondentes pixel a pixel da imagem original, sendo denominadas como mapas de atributos ou *feature-maps* SOUZA, ARAUJO, SILVA e SANTOS (2020). Outro ponto importante, é que esses mapas armazenam também os pesos de conexões entre os neurônios, proporcionando assim a sua aplicação a diferentes pontos da imagem.

Para definir a aplicação destas funções em diferentes camadas da rede, combinada com os pesos de conexão entre os neurônios, possibilita que a rede neural processe informações e aprenda representações complexas dos dados de entrada. A utilização de ReLU (Rectified Linear Unit) nas camadas ocultas introduz

não linearidades, enquanto o Softmax na camada de saída fornece uma interpretação probabilística das previsões da rede. Essas características são essenciais para que a rede possa aprender a mapear padrões nos dados e realizar reconhecimento de padrões em imagens.

Em definição podemos classificar *ReLU* como uma função de ativação não linear que transforma os valores de entrada, se o valor de entrada for positivo, ele permanece inalterado; se for negativo, é substituído por zero. Essa simplicidade e não-linearidade ajudam na convergência eficiente do treinamento da rede neural.

Enquanto Softmax é usada na camada de saída de uma rede neural quando estamos lidando com problemas de classificação multiclasse, transformando um vetor de valores reais em um vetor de probabilidades, onde a soma das probabilidades é igual a 1.

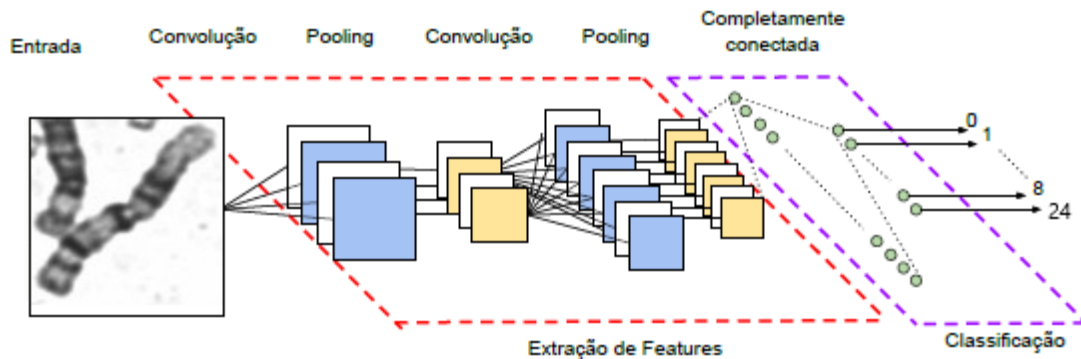
Em paralelo temos o processo de Flatten, onde a operação de converter uma matriz multidimensional em uma única dimensão, ou seja, achatá-la. Isso é frequentemente necessário ao passar dados de uma camada para outra, especialmente ao conectar camadas totalmente conectadas.

Já a camada de agrupamento, quando recebe os dados na forma de matrizes, gerados pelas camadas convolucionais, opera em formas de redução da quantidade de dados. Tendo em vista o contexto de matrizes, tais processos serão aplicados para promover reduções de linhas e colunas, ou ainda, de largura e altura. Convencionalmente, são utilizadas técnicas de *pooling* e *downsampling* para agrupar os principais dados recebidos, sendo os mais comuns o *Max-pooling*, que simplificadaamente aplica uma forma de máscara nos dados dispostos em linhas e colunas dos *feature-maps*, gerando uma nova matriz resultante, montada com os valores máximos identificados na representação do “pixel” central, e o *Average-pooling*, que segue o mesmo passo de aplicação de uma máscara com um percorrimeto de vizinhança de pixels da matriz, mas que obtém o valor médio entre os pixels fornecidos. Com a utilização destas técnicas de redução, consegue-se manter os dados mais importantes de representação de objetos ou composições da imagem, sem causar grande impacto nos processos de identificação posteriores.

Por fim, logo após a(s) camada(s) de agrupamento torna-se necessário aplicar ao menos uma camada de neurônios totalmente conectados, pelo fato de

proporcionar uma grande quantidade de caminhos de tomada de decisão à rede convolucional. Seguindo a mesma construção, a última camada da RNC também é composta por uma densa quantidade de neurônios, agora pelo fato de que essa será de fato responsável pelo processo de identificação dentro da Rede Conexionista de camadas convolucionais.

Figura 11: Representação simplificada de uma estrutura de rede convolucional



Fonte: Adaptada de VARGAS, PAES e VASCONCELOS (2008)

No modelo de rede convolucional apresentado na Figura 11, temos representadas as distribuições de camadas de sua composição, sendo separadas em basicamente dois grupos, um de Extração dos mapas de características e outro de Classificação. Vale destacar que em aplicações práticas, são distribuídas várias camadas de convolução e pooling, seguidas por camadas densas de neurônios de classificação. No exemplo fornecido, é utilizada uma imagem de cromossomo da classe A1.

2.5.1 Modelos de redes convolucionais utilizadas

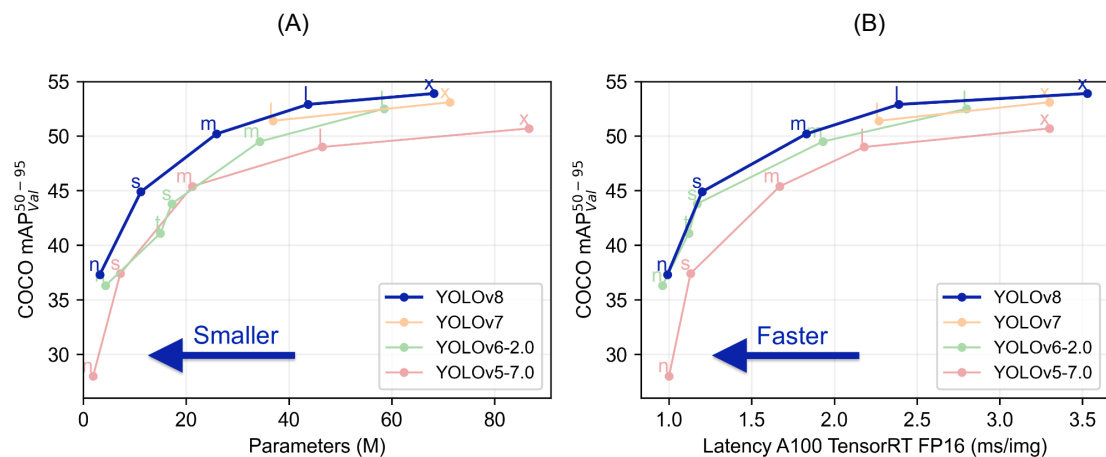
Dado o objetivo do trabalho, tendo em vista o foco na comparação de bibliotecas de Redes Neurais Convolucionais para classificação de imagens, fez-se necessária a utilização de um conjunto de rede que se mostrasse adequado para a realização do processo. Diante os impasses foram selecionadas as seguintes redes de detecção de objetos em imagens: **YOLOV8**, **VGG16**.

2.6 Método Yolo (You Only Look Once)

Refere-se a um Framework de detecção de objetos em imagens e vídeos que se destacam pela eficiência e rapidez. Desenvolvido para otimizar o processo de detecção em tempo real, o YOLO trata o problema de detecção de objetos como uma única tarefa de regressão de caixa delimitadora e probabilidade de classe.

Ao contrário dos métodos tradicionais que dividem a imagem em múltiplas regiões e as examinam separadamente, o YOLO divide a imagem em uma grade e prevê a caixa delimitadora (bounding boxes) e a probabilidade de classe para cada célula da grade. Isso permite que o modelo faça previsões em uma única passagem pela rede neural, permitindo uma detecção mais rápida.

Gráfico 1 – Comparativo YOLOv8 com as versões anteriores



Fonte: GITHUB ULTRALYTICS

O Gráfico 1 mostra uma comparação do YOLOv8 proposto pelos autores com as versões anteriores. O YOLOv8 se mostra a frente das suas versões sendo mais fácil de se utilizar e mais rápido em sua execução, tem melhorias no mAP (mean Average Precision) e FPS (Frames per Second) quando comparado ao YOLOv7.

2.7 Método VGG16

Esta é uma arquitetura de rede neural convolucional (CNN) desenvolvida pelo Visual Geometry Group (VGG) da Universidade de Oxford. Proposto por SIMONYAN

E ZISSERMAN (2014). O nome VGG16 refere-se à profundidade de 16 camadas, incluindo 13 camadas convolucionais e 3 camadas totalmente conectadas.

Foi amplamente reconhecido pela sua eficácia no reconhecimento de imagens. Com sua arquitetura profunda e estrutura simples, o VGG16 tem um bom desempenho em competições de classificação de imagens, especialmente quando treinado em grandes conjuntos de dados como o ImageNet.

2.8 Dataset

Certamente, a tarefa de reconhecimento de cromossomos apresenta alguns desafios, como a determinação de qual processo de divisão celular se tornam melhor identificáveis ou ainda no acerto da precisão de identificação dos mesmos na etapa de segmentação e detecção de características. Partindo do ponto que tais considerações já foram resolvidas e que foram obtidos os dados para reconhecimento, temos agora o fato de que algumas dessas imagens cromossômicas apresentam características variadas, que podem dificultar os processos seguintes de identificação, como variação de coloração, ruídos perceptíveis, diferenças de contraste, etc.

Além disso, outro desafio é a grande quantidade de dados gerados pelas análises cromossômicas. A análise manual demanda muito tempo e esforço de especialistas treinados, o que abre espaço para diferentes interpretações dos resultados, tornando a automação do processo por meio de algoritmos de reconhecimento uma forma viável de aplicação, reduzindo erros e aumentando a eficiência em alguns pontos. Outro ponto relevante, seria a qualidade geral das imagens obtidas no que tange a resolução, visto que imagens com baixa qualidade influenciam diretamente nos processos de treinamento de uma rede convolucional.

Como forma de auxiliar no contorno desses possíveis problemas, foi selecionado um dataset previamente modificado, possuindo em sua totalidade, imagens já convertidas para uma mesma escala de cinza, e com resoluções satisfatórias para um processo de reconhecimento. Outro fato, é que sua disponibilidade original se caracterizava com uma divisão da classificação dos diferentes tipos de dados armazenados, dispondo-se em 24 classes diferentes de

cromossomos, estabelecidas pelas características genéticas singulares, atribuídas com base nas identificações laboratoriais. Foi justamente por esse fator que tal base de dados foi utilizada, pois o processo de divisão e rotulação prévia dos dados já haviam sido realizados.

Figura 12: Imagem 103064 do dataset utilizado



Fonte: Autor (2023)

Sobre seu processo de construção, foram descritas as seguintes informações:

Processo de captura das imagens: Foi utilizada uma câmera montada acima do microscópio, onde os especialistas envolvidos, através de ajustes de lente e ângulos para detecção de coloração, capturaram as imagens e as transferiram para o computador, para posterior processamento.

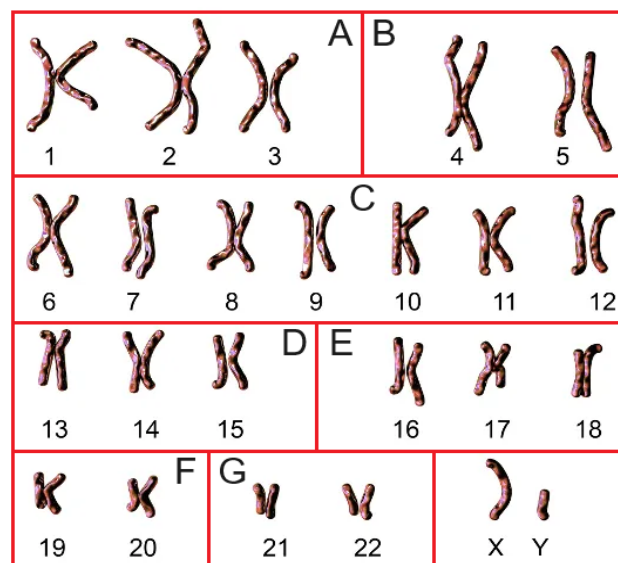
Detalhes técnicos: No estudo base, foi utilizada a técnica de colheita *in situ*, com procedimentos de coloração tripsina/Wright para preparação da banda. As células amnióticas base utilizadas foram cultivadas em meio BIO-AMF, por 6 a 8 dias. Posteriormente, foram tratadas por colcemida por cerca de 30 minutos, para que pudessem ser detidas na fase metafásica. Por fim, as células foram inchadas com uma solução hipotônica e fixadas com uma mistura de metanol/ácido acético; foram tratadas com tripsina e depois coradas com solução de coloração de Wright.

Vale destacar que, para cada imagem contida no dataset, haviam identificações prévias, descritas através de anotações em arquivos xml, dos determinados ‘objetos’ de cromossomos das classes distintas. Dessa forma, alguns processos de ajuste foram necessários, para adaptar os datasets que seriam fornecidos de base para cada modelo, levando em consideração que os parâmetros de entradas de dados se mostraram diferentes para cada rede convolucional, tendo como similaridade somente o fato de que, para cada estrutura, todas as imagens fornecidas deveriam ter as mesmas dimensões de largura e altura.

2.9 Classes dos cromossomos identificados

Na Figura 13, é ilustrada a sequência base de identificações das classes a serem fornecidas para o treinamento dos modelos convolucionais, sendo que as identificações são determinadas pela letra da classe, somada a sua identificação numérica.

Figura 13: Representação das classes de cromossomos utilizados.



Fonte: BiologiaNet (2023)

As classes são divididas em 24, e separadas por grupos e nomeadas de acordo com sua classificação numérica.

3 DESENVOLVIMENTO

O processo de desenvolvimento e a implementação dos algoritmos foram adicionadas como repositório no github, para facilitar a visualização do projeto, disponível em: <https://github.com/Nathan-Nicolau/TCC-Algoritmos>.

3.1 Ambientes de Execução

Referente ao desenvolvimento, ambos os algoritmos foram testados em dois ambientes diferentes, viabilizando a comparação de acurácia entre máquinas com desempenho e hardwares distintos.

Máquina 1:

Processador: Intel core i3 10ª geração
Núcleos: 4 Threads: 8
Clock base: 3600 Mhz (3.6 Ghz), Clock Boost 3900 Mhz (3.9Ghz)
Cache L3: 4MB
GPU: Nvidia GeForce GTX 1060 3 GB de VRAM
Memória RAM: 16 GB

Máquina 2:

Processador: AMD Ryzen 5 2400g
Núcleos: 4 Threads: 8
Clock base: 3600 Mhz (3.6 Ghz), Clock Boost 3900 Mhz (3.9Ghz)
Cache L3: 4MB
GPU AMD Radeon RX 580 8 GB de VRAM
Memória RAM: 16 GB

IDE Utilizada para o modelo de YOLOv8:

PyCharm Community, versão 232.10227.11

Pytorch, versão 2.0.1+cu117

IDE Utilizada para o modelo VGG16:

PyCharm Community, versão 232.10227.11

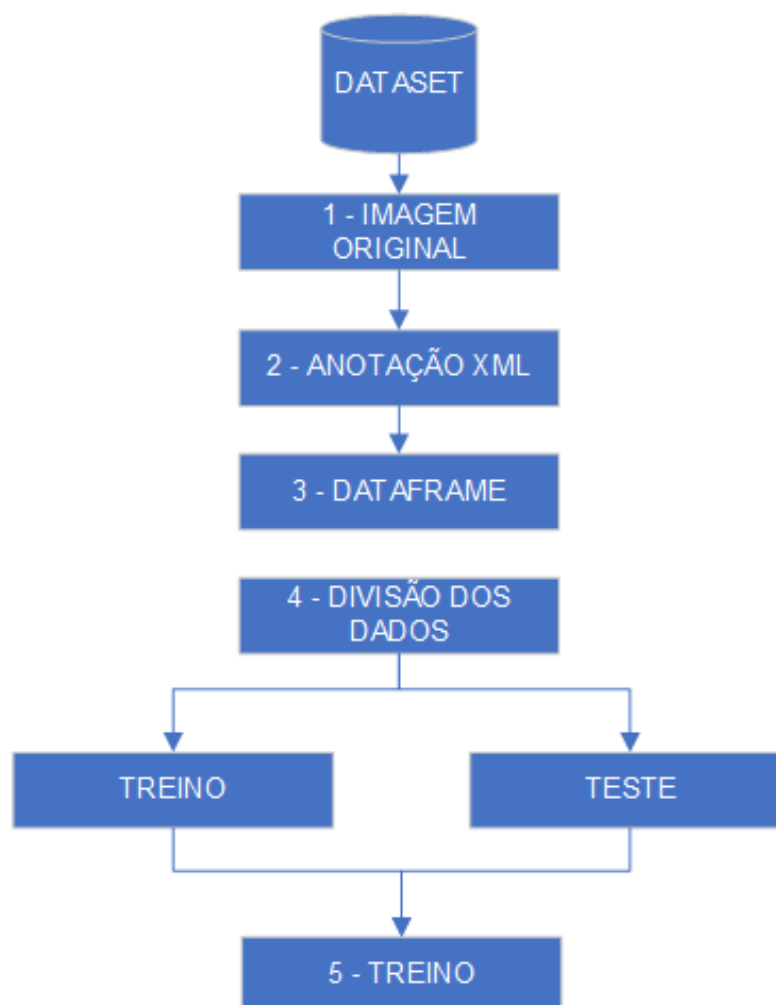
Biblioteca de Inteligência Artificial : Keras versão 2.5

Os processos de ajuste descritos a seguir descrevem alguns passos que se mostraram necessários para que o dataset base fosse igualmente utilizado por ambas as estruturas, sendo distinguidos por somente algumas etapas diferentes.

3.2 Processo de ajuste do dataset para a solução VGG16

O fluxograma descrito na Figura 14 , serve como forma de visualização dos processos de ajustes necessários dos dados (imagens) para o modelo **VGG16**:

Figura 14: Organograma que apresenta fluxo de processos no ajuste das imagens do dataset original



Fonte: Autor (2023)

Onde, são percorridas as seguintes etapas:

- **Etapa 1:** A imagem original do dataset é fornecida para a montagem dos dados de treinamento da rede.
- **Etapa 2:** Junto à imagem, é fornecido seu arquivo de anotação XML, contendo as informações necessárias para identificação dos cromossomos presentes, como classe pertencente, dimensões de largura e altura e posições nos eixos x e y.
- **Etapa 3:** Através da utilização da biblioteca pandas, é montado um DataFrame, contendo uma relação da imagem que vai ser fornecida e um dicionário de dados contendo as informações da Etapa 2.
- **Etapa 4:** Através do uso da função *train_test_split* da biblioteca do scikit-learn, são divididos os dados usados para treinamento e validação da rede.
- **Etapa 5:** Por fim, é realizado o processo de treinamento do modelo de rede convolucional, com os devidos ajustes dos hiperparâmetros relacionados, como quantidade de imagens por iteração da rede (batch_size) ou ainda a quantidade de épocas (epochs) determinadas.

3.3 Processo de treinamento do modelo VGG16

Os hiperparâmetros configurados para o treinamento do modelo foram:

- **Dimensão (input_shape):** 224 x 224
Dimensão padronizada das imagens que foram fornecidas para treinamento. Valor padrão melhor estabelecido para esse modelo de rede convolucional
- **Número de épocas (epochs):** 100
Quantidade definida numa média calculada entre ambas as redes, para melhores resultados possíveis de loss e accuracy.

- **Tamanho do lote (batch_size): 64**
Valor médio de fornecimento entre modelos previamente testados, que obteve melhores resultados.
- **Taxa de aprendizado (learning_rate): 0.0001**
Valor padrão estabelecido da *taxa de aprendizado* utilizada na maior parte dos modelos de aprendizado supervisionado, responsável pelo ajuste dos pesos.
- **Parada antecipada (early_stopping):** monitor='val_accuracy', patience=5, restore_best_weights=True
Parâmetro utilizado para monitorar os valores apresentados no aprendizado da rede, promovendo uma interrupção prévia do modelo, quando o valor monitorado apresenta uma constância de imutabilidade dentro do escopo de épocas determinado no seu sub-parâmetro *patience*.
- **Quantidade total de imagens de treinamento: 5000**
Quantidade total das imagens presentes no dataset utilizado. Representa uma quantidade significativamente boa para os objetivos previamente propostos.

Tabela 1: Representação dos dados da primeira época de treinamento

Epoch 1			
Epochs/steps	Time(seconds)	loss	accuracy
1/60	16:24	4.1086	0.0000e+00
10/60	11:29	1.1116	0.7633
20/60	09:41	0.5385	0.8855
30/60	07:28	0.3552	0.9245
40/60	05:04	0.2650	0.9437
50/60	02:33	0.2113	0.9551
60/60	19s	0.1757	0.9626

Fonte: Autor (2023)

Durante a primeira época de treinamento do modelo VGG16, foram observados diversos aspectos reveladores do comportamento do modelo ao longo do processo de aprendizado. A análise detalhada das diferentes etapas,

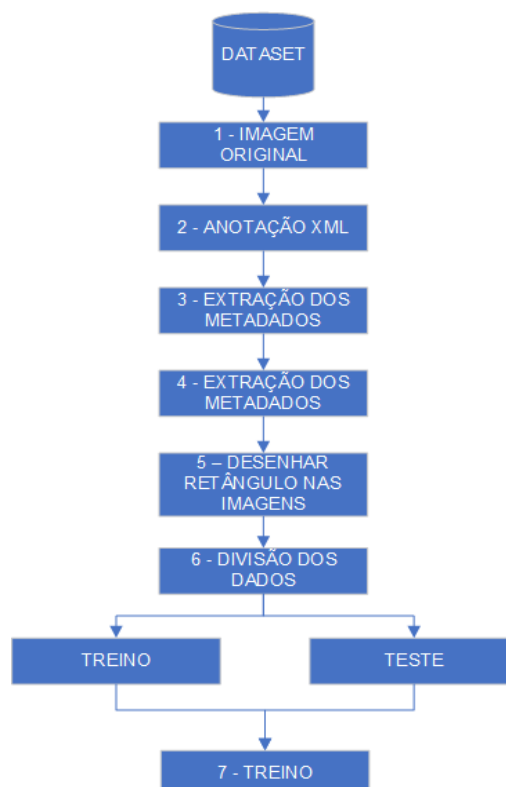
representadas por batches, proporciona uma compreensão mais aprofundada do desempenho inicial do modelo.

Ao longo dos batches, observamos uma progressão notável na redução da loss e um aumento consistente na acurácia, indicando que o modelo está se ajustando e aprendendo padrões nos dados de treinamento. A acurácia final atingiu 96.26% no último batch desta época, sugerindo uma adaptação inicial bem-sucedida aos dados. Esses resultados fornecem uma visão inicial encorajadora do potencial de aprendizado do modelo VGG16 neste contexto específico do modelo.

3.4 Processo de ajuste do dataset para o modelo YOLOv8

As entradas de dados para o modelo foram as próprias imagens laboratoriais do dataset original, sendo que o resultado desejado é a identificação dos pares de cromossomos presentes na imagem. O fluxo de trabalho para alcançar esse objetivo é composto pelos elementos identificados no fluxograma da Figura 15:

Figura 15: Diagrama sobre o fluxo do modelo Yolo



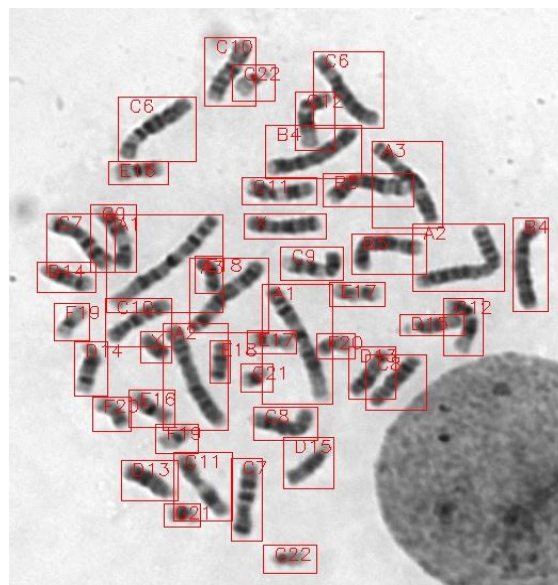
Fonte: Autor(2023)

É importante mencionar que todos os passos devem ser feitos corretamente para que o treinamento da rede possa ser realizado.

3.5 Processo de classificação das imagens para o modelo YOLOv8

Com base no dataset que foi escolhido para este projeto, o mesmo conta com as anotações das imagens dentro de arquivos XML no qual tem informações sobre o nome da imagem, os objetos presentes na imagem junto a sua localização e sua classe, assim, para cada objeto da imagem, foi desenhado um retângulo para identificar o objeto e dentro do mesmo, foi inserido um texto para identificar a sua classe, devido a isso, o nosso trabalho reduziu drasticamente, pois foi implementado um algoritmo python para extrair as informações desses arquivos XML, desenha re inserir os retângulos, inserir a legenda e salvar nas pastas de treino e validação, que contém duas sub-pastas, imagens e anotações, para este projeto foi utilizado um total de 5000 imagens que estão presentes no dataset, dividimos os nossos dados respeitando a proporção de 75% dados para treinamento e 25% dados para validação. A Figura 16 tem como objetivo visualizar as imagens depois de finalizado o processo de classificação, respeitando o formato padrão do método YOLOv8.

Figura 16: Imagem após processo de delimitação supervisionada



Fonte: Autor (2023)

3.6 Processo de treinamento do modelo YOLOv8

Com a base de dados pronta, o treinamento foi realizado utilizando a IDE Pycharm, as figuras mostram o resultado do treinamento a partir do seu hardware e placa gráfica específica, para este treinamento foram utilizados os seguintes hiperparâmetros:

- **Dimensão das imagens (input_shape): 640:**
Dimensão de entrada inicial que o Yolo pede como configuração padrão na documentação, assim mantivemos o tamanho dimensionado de 640x640.
- **Tamanho do lote (batch_size): 4**
Valor médio de fornecimento entre modelos previamente testados, que obteve melhores resultados
- **Número de Épocas (epochs): 100.**
Quantidade definida numa média calculada entre ambas as redes, para melhores resultados possíveis de loss e accuracy.

3.6.1 Treinamento feito na Máquina 1:

Tempo de processamento:

Para a execução das 100 épocas, dentro do dataset completo, o processo de execução passou por todos os treinamentos em aproximadamente 4 horas. Dentro desse período a média de resultados no treinamento pode ser visualizada na Tabela 2:

Tabela 2: Resultados obtidos do modelo Yolov8 treinado em um ambiente com GPU

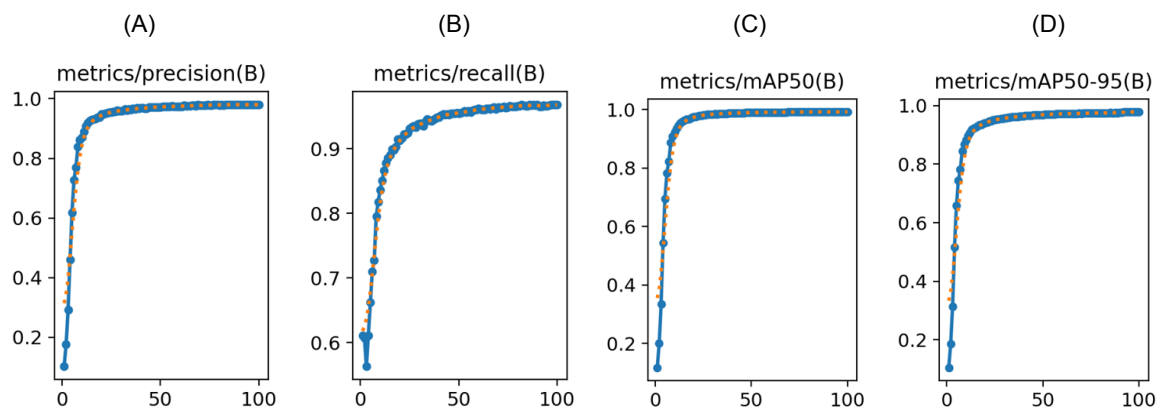
Class	Images	Instances	Box Precision	Box Recall	Box mAP50	Box mAP50-95
ALL	1250	57487	0.981	0.967	0.992	0.979
A1	1250	2499	0.979	0.986	0.994	0.991
A2	1250	2500	0.980	0.978	0.993	0.99
A3	1250	2500	0.982	0.969	0.993	0.988
B4	1250	2501	0.987	0.971	0.994	0.989
B5	1250	2500	0.981	0.974	0.993	0.988
C6	1250	2500	0.970	0.956	0.99	0.985
C7	1250	2501	0.98	0.977	0.994	0.987
C8	1250	2500	0.983	0.956	0.992	0.985
C9	1250	2499	0.98	0.951	0.991	0.981
C10	1250	2499	0.974	0.965	0.992	0.984
C11	1250	2499	0.986	0.96	0.992	0.984
C12	1250	2500	0.982	0.97	0.993	0.985
D13	1250	2499	0.969	0.933	0.988	0.977
D14	1250	2501	0.984	0.963	0.993	0.98
D15	1250	2499	0.955	0.943	0.987	0.973
E16	1250	2496	0.976	0.955	0.991	0.976
E17	1250	2497	0.988	0.977	0.994	0.976
E18	1250	2503	0.975	0.968	0.993	0.975
F19	1250	2499	0.985	0.976	0.994	0.969
F20	1250	2500	0.99	0.987	0.995	0.971
G21	1250	2504	0.991	0.968	0.994	0.955
G22	1250	2496	0.98	0.985	0.993	0.959
X	1250	1843	0.993	0.976	0.993	0.986
Y	1250	652	0.991	0.977	0.991	0.955

Fonte: Autor (2023)

A Tabela 2 tem como objetivo exibir os resultados durante o processo de treinamento, onde a acurácia é delimitada pelo último parâmetro que determina em

porcentagem a precisão durante a análise e identificação dos cromossomos e suas classes, sendo em comparação de 0 para 1, sendo 1 = 100%.

Gráfico 2 - Resultados de treinamento do YOLOv8 na máquina 1



Fonte: Autor (2023)

Observando o Gráfico 2, nota-se que o modelo atingiu as expectativas no treinamento, sendo relevante considerar as informações de *precision*, *recall*, mAP50, mAP50-95, onde o nível de confiança de precisão e revogação durante o treinamento e validação se mostraram promissores, atingindo 98%, o que representa uma boa precisão.

3.6.2 Treinamento feito na Máquina 2:

Ajustando os mesmos parâmetros do primeiro teste, foi levantado na máquina com a arquitetura AMD, a comparação do desempenho e após a execução do processo de treinamento. Após a finalização podemos denotar os seguintes aspectos:

Tempo de processamento:

Para a execução das 100 épocas, dentro do dataset completo, o processo de execução passou por todos os treinamentos em aproximadamente 56 horas. Dentro desse período a média de resultados no treinamento pode ser visualizada na Tabela 3:

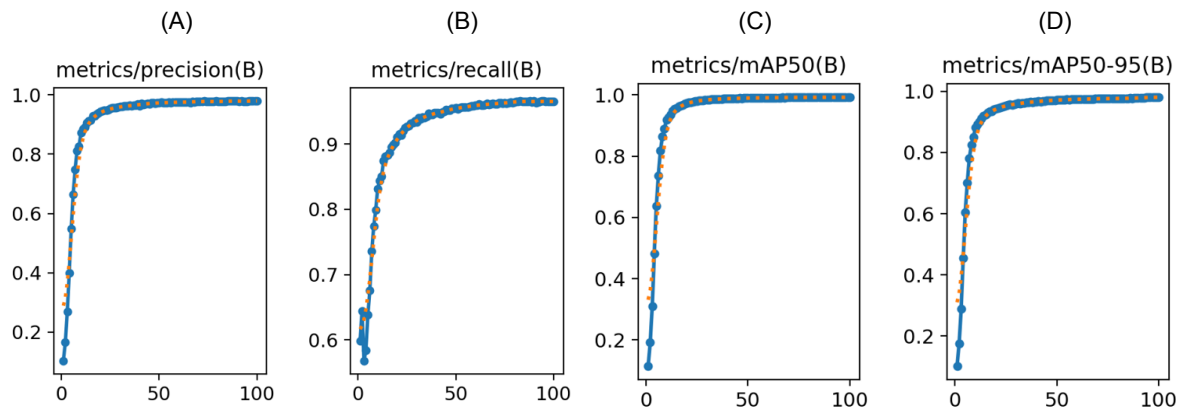
Tabela 3: Resultados obtidos do modelo YOLOv8 treinado em um ambiente com CPU

Class	Images	Instances	Box Precision	Box Recall	Box mAP50	Box mAP50-95
ALL	1250	57487	0.981	0.965	0.992	0.981
A1	1250	2499	0.976	0.984	0.994	0.991
A2	1250	2500	0.981	0.979	0.994	0.99
A3	1250	2500	0.985	0.963	0.992	0.989
B4	1250	2501	0.987	0.974	0.994	0.99
B5	1250	2500	0.982	0.972	0.994	0.989
C6	1250	2500	0.966	0.95	0.989	0.984
C7	1250	2501	0.982	0.97	0.993	0.987
C8	1250	2500	0.982	0.958	0.992	0.987
C9	1250	2499	0.976	0.945	0.990	0.983
C10	1250	2499	0.974	0.960	0.992	0.985
C11	1250	2499	0.985	0.957	0.992	0.986
C12	1250	2500	0.983	0.970	0.993	0.986
D13	1250	2499	0.966	0.932	0.987	0.977
D14	1250	2501	0.983	0.962	0.993	0.982
D15	1250	2499	0.950	0.941	0.986	0.975
E16	1250	2496	0.979	0.955	0.991	0.979
E17	1250	2497	0.987	0.974	0.993	0.980
E18	1250	2503	0.981	0.966	0.993	0.979
F19	1250	2499	0.984	0.977	0.994	0.972
F20	1250	2500	0.992	0.986	0.995	0.974
G21	1250	2504	0.989	0.965	0.994	0.962
G22	1250	2496	0.980	0.978	0.993	0.962
X	1250	1843	0.995	0.971	0.994	0.988
Y	1250	652	0.994	0.975	0.990	0.959

Fonte: Autor(2023)

Após a finalização do processo de treinamento, o Gráfico 3 a seguir ilustra detalhes e dados de crescimento de precisão das métricas aplicadas na sequência de épocas de treinamento do YOLOv8 na máquina 2:

Gráfico 3 - Resultados do treinamento YOLOV8 na Máquina 2



Fonte: Autor (2023)

Analisando o Gráfico 3, nota-se o crescente aumento na taxa de precisão, conforme novas entradas de imagens são recebidas a precisão aumenta de forma gradativa. Iniciou com apenas 10% de precisão e ao final se estabilizou em 99,2%.

4 ANÁLISE DOS RESULTADOS OBTIDOS

4.1 Resultados obtidos no modelo YOLOv8

De acordo com os resultados obtidos no treinamento do modelo YOLOv8, entre ambas as máquinas, foram construídas as seguintes tabelas de resultado:

Tabela 4: Ambiente de execução em GPU - Máquina 1

Class	Images	Instances	Box Precision	Box Recall	Box mAP50	Box mAP50-95
Todas	1250	57487	0.981	0.967	0.992	0.979

Fonte: Autor (2023)

Tabela 5: Ambiente de execução em CPU - Máquina 2

Class	Images	Instances	Box Precision	Box Recall	Box mAP50	Box mAP50-95
Todas	1250	57487	0.981	0.965	0.992	0.981

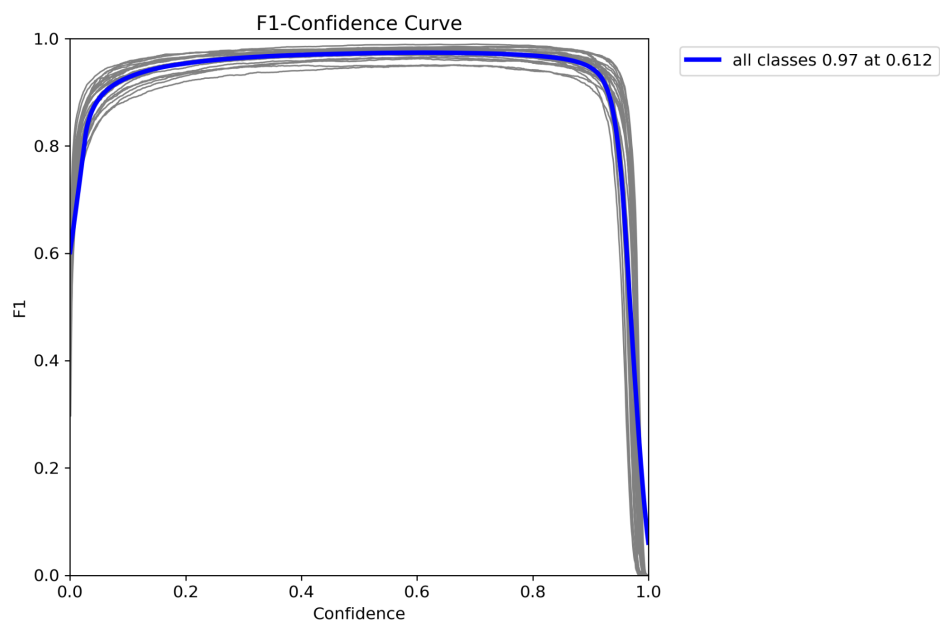
Fonte: Autor (2023)

A partir dos resultados das Tabelas 3 e 4, o treinamento em GPU e CPU se mostram equivalentes, tendo como real diferença o tempo de treinamento entre ambas, devido a GPU Nvidia ter o suporte ao CUDA, utilizado como padrão para execução do YOLO.

Informações médias obtidas :

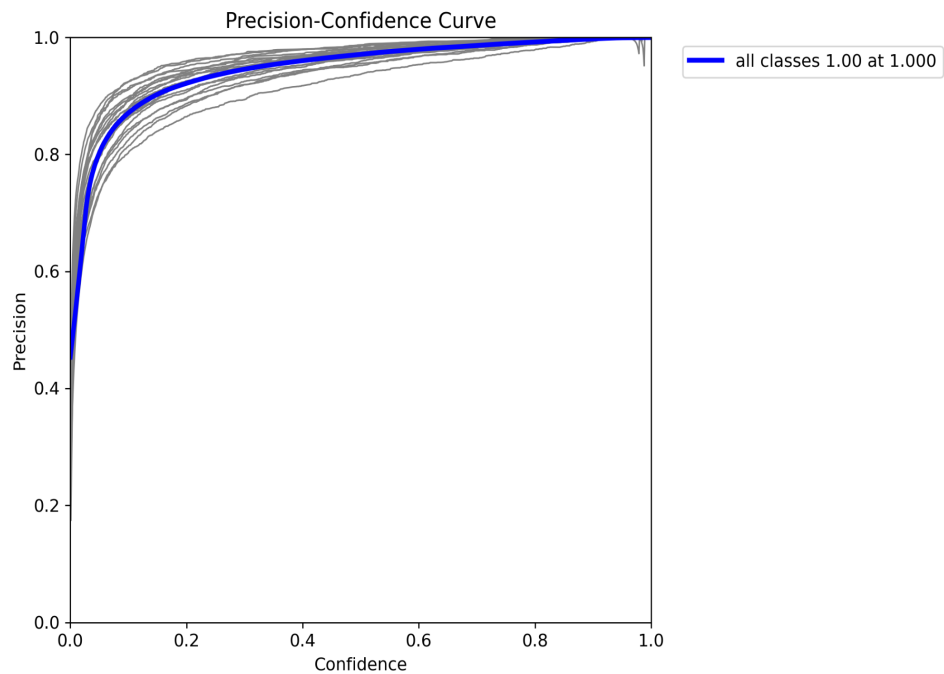
- **Box Precision** (Precisão das Caixas): Em média, as caixas delimitadoras preditas têm alta precisão de 98%.
- **Box Recall** (Recall das Caixas): Indica que o modelo está recuperando corretamente cerca de 96% das caixas delimitadoras de referência.
- **Box mAP50** (média da Precisão da Caixa com IoU de 0.5): É uma métrica robusta indicando alta precisão de 99%.
- **Box mAP50-95** (média da Precisão da Caixa com IoU de 0.5 a 0.95): Indica uma precisão sólida em uma variedade de sobreposições de caixas com 98%.

Gráfico 4 - Resultados da métrica F1 que combina precisão (precision) e revocação (recall) em um único valor.



Fonte: Autor (2023)

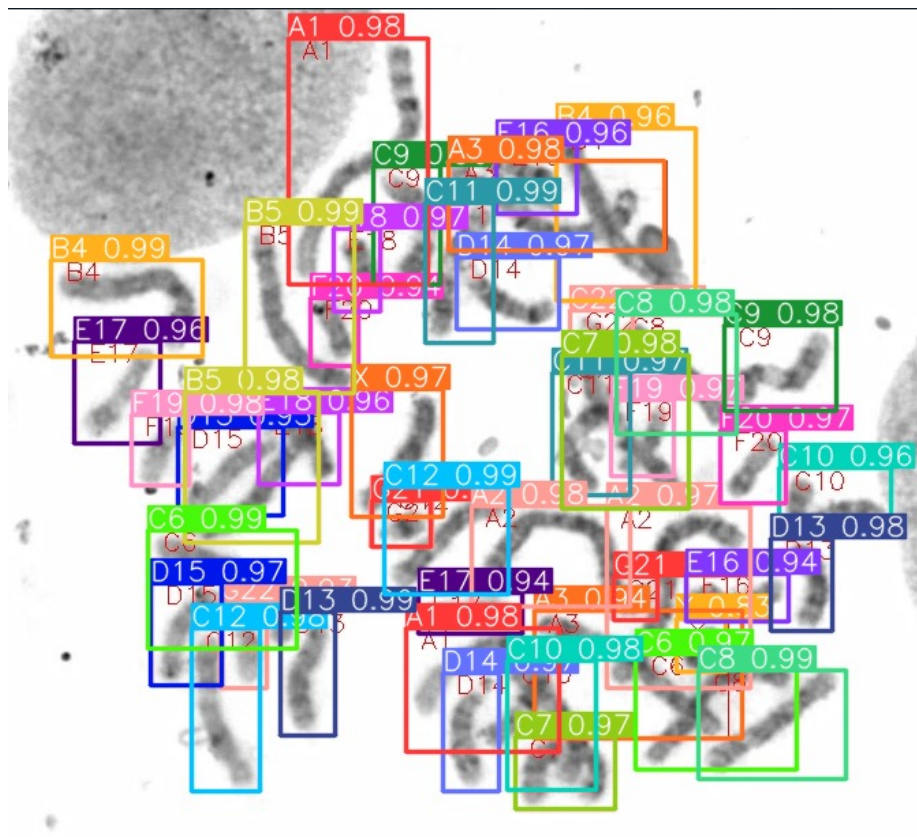
Gráfico 5 - Métrica de curva de precisão e confiança, varia em relação à confiança associada das Previsões



Fonte: Autor (2023)

De acordo com a saída "all classes 0.97 at 0.612" do Gráfico 4, é sugerida que a confiança da detecção está sendo fornecida para todas as classes consideradas pelo modelo. O valor "0.97" representa a confiança, indicando quão certo o modelo está. Uma confiança de 0.97 é relativamente alta, indicando uma detecção bastante confiável. A parte "até 0.612" refere-se à posição da detecção na imagem, o valor "0.612" indica a localização aproximada da detecção ao longo de um dos eixos, seja x ou y. Isso fornece informações valiosas sobre a localização espacial da detecção na imagem. O Gráfico 5 ilustra a curva de precisão e confiança do modelo, isso indica que o modelo está atribuindo uma confiança máxima de 1.00 (ou 100%) à detecção, e a detecção ocorreu na posição 1.000 em um sistema de coordenadas normalizado. Isso sugere que, de acordo com o modelo, a detecção é considerada extremamente confiável, pois a confiança é máxima (1.00). Além disso, a posição "1.000" indica a localização da detecção na imagem, com 1.000 sendo o valor máximo em um sistema normalizado.

Figura 17 - Resultado de detecção em uma nova imagem que não foi utilizada no treinamento

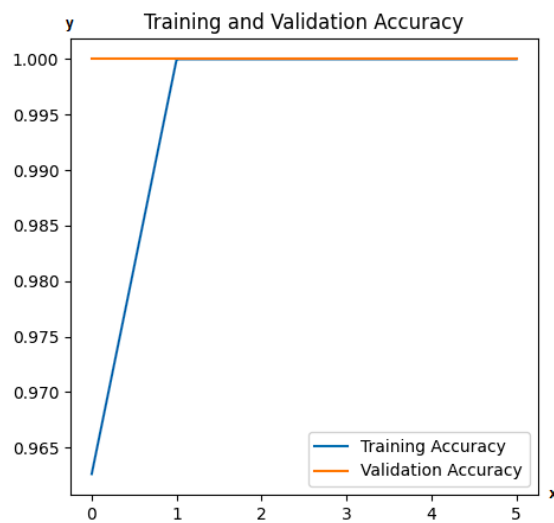


Fonte:Autor (2023)

Como apresentado, o treinamento utilizado como base de conhecimento, foi aplicado em uma nova imagem, onde o modelo se mostra promissor ao relacionar e delimitar os pares de cromossomos a partir de bounding boxes denominadas de acordo com as suas respectivas classes.

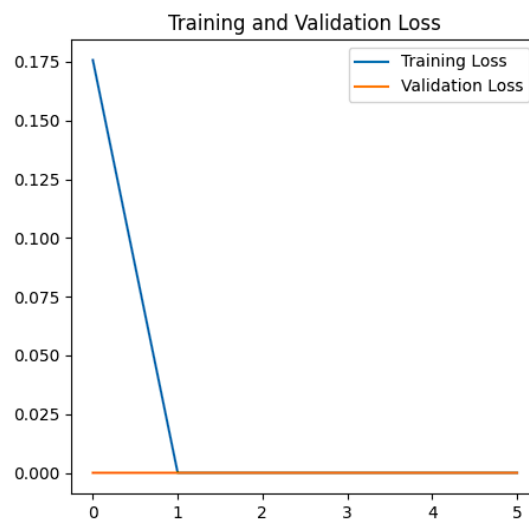
4.2 Resultados obtidos com o modelo VGG16

Gráfico 6 - Porcentagem de acurácia no treinamento e validação das imagens



Fonte: Autor (2023)

Gráfico 7 - Porcentagem de perda das imagens de treinamento e validação



Fonte: Autor (2023)

Nota-se que no Gráfico 6, a reta de cor azul, representando a média de valor na precisão de treinamento em porcentagem - eixo y -, sofre uma crescente exponencial de valor à medida em que o número de épocas - eixo x - cresce, mais especificamente a partir da época 1, mantendo um valor constante de 1.000, ou seja, de 100% em acurácia. Vale mencionar que, a quantidade 5 no número de épocas registradas, se deve ao sub parâmetro *patience* estabelecido no **early_stopping**, pelo fato de que a métrica de acurácia foi monitorada pela quantidade de 5 épocas de inalteração, fazendo com que o processo de treinamento da rede fosse interrompido, evitando um possível *overfitting*. Já a reta de cor laranja, representando a média de valor na precisão de validação em porcentagem - eixo x -, desde a primeira época registrada, apresenta o valor de 1.000, ou seja, de 100% na precisão dos dados de validação, se mostrando inalterada até a última época informada.

O Gráfico 7, por sua vez, fornece uma representação praticamente espelhada no eixo x do Gráfico 6, isso pelo fato das similaridades inversas dos valores, tendo a reta de cor azul se iniciando com uma média de perda no valor de cerca de 17% - eixo x - na primeira época, mas que, na medida que o número de épocas aumenta - eixo y -, seu valor é reduzido a praticamente 0% de perda.

Tabela 6 Representação dos resultados obtidos por época

Epoch	Time(seconds)	loss	accuracy	val_loss	val_accuracy
1	1156s	1.757	9.626	2.5210	1.0000
2	1115s	7.0878e-06	1.0000	6.8276e-07	1.0000
3	1150s	3.4014e-06	1.0000	2.1985e-07	1.0000
4	1089s	2.7047e-06	1.0000	6.8639e-08	1.0000
5	1121s	1.7634e-06	1.0000	7.6545e-09	1.0000
6	1121s	9.0196e-07	1.0000	1.2548e-10	1.0000

Fonte: Autor (2023)

Analisando os dados da Tabela 5, que complementam os Gráficos 6 e 7, nota-se que para 100 épocas, o comando **early_stopping** não identificou melhoria no algoritmo e interrompeu o treinamento na época 6.

Observou-se também um comportamento notável ao longo das diferentes épocas. Cada época representa uma iteração completa pelos dados de treinamento. Na primeira época, observou-se uma acurácia relativamente baixa nos dados de treinamento (9.626%), indicando que o modelo estava inicialmente ajustando-se aos dados. No entanto, a *val_accuracy* atingiu 100.00%, o que pode ser um sinal de *overfitting* (treinamento excessivo), especialmente considerando a diferença entre a acurácia nos dados de treinamento e nos dados de validação.

Nas épocas subsequentes, observa-se uma tendência notável de *overfitting*, pois a acurácia nos dados de treinamento atinge a perfeição (100.00%), enquanto os valores de *val_loss* e *val_accuracy* também atingem valores extremamente baixos, sugerindo uma adaptação excessiva aos dados de treinamento específicos. Essa discrepância entre os resultados nos dados de treinamento e nos dados de validação é um sinal de *overfitting*.

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

No trabalho de conclusão apresentado, explorou-se a aplicação de modelos YOLOv8 e VGG16 em detecção de cromossomos, com o objetivo de compreender a capacidade desses modelos em identificar e catalogar estruturas cromossômicas. A pesquisa buscou investigar se redes neurais convolucionais seriam eficientes nesse contexto. A contribuição teórica deste estudo concentra-se na análise de métodos utilizados para a identificação de objetos em imagens relacionadas à genética. Adicionalmente, o trabalho propõe e implementa um modelo prático de resolução do problema, utilizando Inteligência Artificial e redes neurais convolucionais. Esse modelo, de implementação básica e simples, pode ser aplicado em sistemas destinados à identificação de cromossomos.

Com base no estudo de análise do modelo YOLOv8, utilizado em ambientes diferentes, conclui-se que a diferença entre os resultados, é unicamente no tempo de execução, onde em arquiteturas utilizando GPU NVIDIA, o modelo possui suporte adequado para execução do processo, já para arquitetura AMD, o modelo não tem suporte para executar pela GPU em ambiente Windows, o que tornou o processo de treinamento relativamente demorado.

Já para o modelo VGG16, para o objetivo de detecção de cromossomos, os resultados sugerem que, embora o modelo inicialmente tenha mostrado uma capacidade de aprendizado positiva na primeira época, a segunda época indica um possível problema de *overfitting*. O modelo está se ajustando demais aos dados de treinamento específicos, resultando em uma acurácia perfeita nos dados de treinamento, mas não generalizando bem para os dados de validação.

Partindo para uma comparação direta entre os dois modelos, pode-se concluir que ambos apresentaram excelentes resultados, quando submetidos aos processos de treinamento devidamente ajustados, sendo a média de precisão do modelo YOLOv8 de cerca de 98%, enquanto que a média do modelo VGG16 situou-se no valor de 100% precisão. De forma clara, é notável que a precisão perfeita do modelo baseado na estrutura VGG16 demonstrou-se a ideal a ser aplicada em objetivos de classificações similares, mas deixando em aberto possibilidades de *overfitting* ou ainda de métricas desbalanceadas, sendo

necessário ajustes precisos nos hiperparâmetros para se obter os resultados corretos.

Para projetos futuros, tendo como conhecimento a análise feita para os dois modelos, aprimorar o conhecimento sobre cálculos e um entendimento mais profundo sobre a arquitetura do VGG16 para que possa de fato mostrar resultados eficientes e confiáveis, de modo a perceber que a rede está aprendendo com os dados fornecidos, e assim, fazer uma comparação concreta sobre o modelo YOLOv8.

Para dar continuidade e aprimorar o projeto, como trabalho futuro, pretende-se implementar a ideia inicial de criar um modelo de rede neural convolucional personalizado capaz de realizar a detecção de cromossomos. Isso seria acompanhado por uma interface gráfica que poderia ser utilizada em diversas áreas aplicadas da genética.

6 BIBLIOGRAFIA

ANDRADE, Maria Fernanda S. Transfer Learning para detecção de cromossomos. 48 Páginas. Centro Regional de Ciências do Nordeste, 2019 (pág. 8).

ANDREW, NG. "AI is the new electricity." Revista Fortune, 2017. Disponível em: <https://www.gsb.stanford.edu/insights/andrew-ng-why-ai-new-electricity>.

ARTERO, Almir Olivette. INTELIGÊNCIA ARTIFICIAL: Teórica e prática. 1ª edição. Editora Livraria da Física, 2009.

CHAVES, T. F.; NICOLAU, L. S. "Citogenética & Cariotipagem Humana." Revista Saúde e Desenvolvimento, [S. l.], v. 4, n. 2, p. 57–66, 2013. Disponível em: <https://www.revistasuninter.com/revistasauade/index.php/saudeDesenvolvimento/article/view/229>. Acesso em: 15 de março de 2023.

Chien-Hsing Lu;Chih-En Kuo;Jenn-Jhy Tseng (2022) CIL:54816, Homo sapiens Linnaeus, 1758, epithelial cell. CIL. Dataset. CIL. Dataset. KARGER (2016). <https://doi.org/doi:10.7295/W9CIL54816>

CIMIRRO, Jean Lucas da Silva. "RECONHECIMENTO DE IMAGENS: USO DO MÉTODO YOLO NO RECONHECIMENTO DE PLACAS DE TRANSITO." Universidade Federal do Pampa, Bagé, 2022. 64 Páginas.

COSTA, Anna Helena Reali. Redes Neurais Artificiais. POLI-PCS/EPUSP, s.d.

GITHUB ULTRALYTICS, disponível em: <https://github.com/ultralytics/ultralytics>

GONÇALVES, André Ricardo. "Redes Neurais Artificiais." 22 Páginas. Originalmente publicado em: www.dca.fee.unicamp.br/~andreric. UNICAMP, 2008. Disponível em: https://andreric.github.io/files/pdfs/redes_neurais.pdf.

HAYKIN, Simon. Redes Neurais: Princípios e Práticas. 2ª Edição. Tradução: Paulo Martins Engel. Porto Alegre; Bookman, 2001.

“Inteligência artificial, aprendizado de máquina, diagnóstico auxiliado por computador e radiônica: avanços da imagem rumo à medicina de precisão.”

Marcel Koenigkam Santos^{1,a}, José Raniery Ferreira Júnior^{2,3,b}, Danilo Tadao Wada^{1,c}, Ariane Priscilla Magalhães Tenório^{3,d}, Marcello Henrique Nogueira-Barbosa^{3,e}, Paulo Mazzoncini de Azevedo Marques^{3,f}

Disponível em:
<https://www.scielo.br/j/rb/a/9yX6w83KDDT33m6G9ddCqBn/?format=pdf&lang=pt>

JURASZEK, Guilherme Defreitas. "RECONHECIMENTO DE PRODUTOS POR IMAGEM UTILIZANDO PALAVRAS VISUAIS E REDES NEURAIS CONVOLUCIONAIS." Joinville, SC. 2014. 155 Páginas.

Livro Digital Deep Learning Book – O Neurônio, Biológico e Matemático.

Disponível em:
<https://www.deeplearningbook.com.br/o-neuronio-biologico-e-matematico/>

Lopes, F. P., & Marana, A. N. "Chromosome identification and counting using mathematical morphology and artificial neural networks." Pattern Recognition Letters (2018), p105, 36-44. Disponível em:
<https://www.sciencedirect.com/science/article/abs/pii/S0167865517303306?via%3Dihub>.

MOREIRA, Fabiano Cordeiro. "RECONHECIMENTO E CLASSIFICAÇÃO DE PADRÕES DE IMAGENS DE NÚCLEOS DE LINFÓCITOS DO SANGUE PERIFÉRICO HUMANO COM A UTILIZAÇÃO DE REDES NEURAIS ARTIFICIAIS."

69 Páginas. Universidade Federal de Santa Catarina, Florianópolis, Fevereiro de 2002.

MUNIZ, Milton. Citogenética. Florianópolis: UFSC, 2009. 126 p.

REIS, Matheus Pevidor. "RECONHECIMENTO DE PESSOAS ATRAVÉS DA BIOMETRIA DA FACE E DA ORELHA." 64 Páginas. Instituto de Ciências Exatas e Biológicas - Departamento de Computação. Universidade Federal de Ouro Preto, MG. Março de 2023.

Santos, J. F., Ferreira, R. A., & Carvalho, T. "Automated chromosome counting in digital images using convolutional neural networks." *Journal of Medical Systems* (2019), pgs 43(3), 48. Disponível em: <https://doi.org/10.1007/s10916-019-1193-3>.

Santos, J. F., & Marana, A. N. "Automated chromosome counting using artificial neural networks." *Expert Systems with Applications* (2019); 125 Páginas, 73-82. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0957417419300478?via%3Dihub>.

SILVA, Ivan Nunes da, SPATTI, Danilo Hernane; FLAUZINO, Rogério Andrade. *Redes Neurais Artificiais PARA ENGENHARIA E CIÊNCIAS APLICADAS: fundamentos teóricos a aspectos práticos*. 2ª Edição revisada e ampliada. Artliber, 2016. Universidade de São Paulo, 2016.

SOUZA, Victor. ARAUJO, Leandro. SILVA, Luan. SANTOS, Adam. "Análise Comparativa de Redes Neurais Convolucionais no Reconhecimento de Cenas." Balneário Camboriú, SC. Apresentado entre 2 e 4 de setembro de 2020.

UNIVESP TV. "Aprendizado de Máquinas - Conceitos Básicos de Aprendizado de Máquina." Disponível em:

https://www.youtube.com/watch?v=sLZeBzFPi10&list=PLhz-BP5S9jxbkQJl5Byf4vnHexSFbDA4I&index=9&ab_channel=UNIVESP. Acesso em 4 de março de 2023.

Definição sobre os cariótipos: <https://www.biologianet.com/genetica/cariotipo.htm>.

SHARMA, M., VIG, S. "Automatic Chromosome Classification using Deep Attention Based Sequence Learning of Chromosome Bands." 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brasil, 2018, pp. 1-8, doi: 10.1109/IJCNN.2018.8489321.

“Você sabe o que são cromossomos?” PUBLICADO POR HOSPITAL ISRAELITA ALBERT EINSTEIN EM 30/03/2022 | Atualizado em 29/01/2023. Disponível em: <https://vidasaudavel.einstein.br/voce-sabe-o-que-sao-cromossomos/#:~:text=Os%20cromossomos%20são%20estruturas%20que,informação%20genética%20herdada%20dos%20progenitores.>