

UNIVERSIDADE FEDERAL DE ALFENAS

GABRIEL THIAGO FERNANDES

GUSTAVO FIDELIS CAMILO

**AUTOMAÇÃO DA ANÁLISE DE CÉLULAS VEGETAIS: UM *PIPELINE* HÍBRIDO
BASEADO EM *CELLPOSE* E *EFFICIENTNET***

ALFENAS/MG

2025

GABRIEL THIAGO FERNANDES
GUSTAVO FIDELIS CAMILO

**AUTOMAÇÃO DA ANÁLISE DE CÉLULAS VEGETAIS: UM *PIPELINE* HÍBRIDO
BASEADO EM *CELLPOSE* E *EFFICIENTNET***

Monografia apresentada como parte dos requisitos para
obtenção do título de Bacharel em Ciência da
Computação pela Universidade Federal de Alfenas.
Área de concentração: Ciências Exatas.

Orientador: Prof. Dr. Luiz Eduardo da Silva

ALFENAS/MG

2025

Sistema de Bibliotecas da Universidade Federal de Alfenas
Biblioteca Unidade Educacional Santa Clara

Fernandes, Gabriel Thiago.

Automação da análise de células vegetais : um pipeline híbrido baseado em cellpose e efficientnet / Gabriel Thiago Fernandes, Gustavo Fidelis Camilo. - Alfenas, MG, 2025.

41 f. : il. -

Orientador(a): Luiz Eduardo da Silva.

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Federal de Alfenas, Alfenas, MG, 2025.
Bibliografia.

1. Citogenética. 2. Deep Learning. 3. Cellpose. 4. EfficientNet. 5. Automação. I. Camilo, Gustavo Fidelis. II. Silva, Luiz Eduardo da, orient. III. Título.

GABRIEL THIAGO FERNANDES
GUSTAVO FIDELIS CAMILO

**AUTOMAÇÃO DA ANÁLISE DE CÉLULAS VEGETAIS: UM *PIPELINE* HÍBRIDO
BASEADO EM *CELLPOSE* E *EFFICIENTNET***

O Presidente da banca examinadora abaixo assina a aprovação da Monografia apresentada como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação pela Universidade Federal de Alfenas. Área de concentração: Ciências Exatas.

Aprovada em: 10 de dezembro de 2025

Prof. Dr. Luiz Eduardo da Silva
Universidade Federal de Alfenas

Assinatura:

Prof.^a Dr.^a Mariane Moreira de Souza
Universidade Federal de Alfenas

Assinatura:

Prof. Dr. Antonio Rodrigues da Cunha Neto
Universidade Federal de Alfenas

Assinatura:

RESUMO

A análise citogenética de células vegetais é uma etapa fundamental em estudos de toxicologia ambiental, genética e fisiologia vegetal. Entretanto, o processo tradicional de identificação das fases mitóticas em lâminas citológicas é manual, demorado e sujeito a variabilidade entre observadores. Este trabalho propõe um *pipeline* automatizado para segmentação e classificação de células em diferentes fases da divisão celular, integrando duas abordagens de aprendizado profundo: o modelo *Cellpose* para segmentação e a arquitetura *EfficientNet* para classificação citológica. O *dataset* utilizado foi composto por 656 imagens microscópicas contendo células de *Allium cepa* e *Lactuca sativa*, das quais foram extraídos originalmente 993 recortes de células, sendo 442 em interfase e as demais distribuídas entre as fases mitóticas, conjunto este que foi posteriormente ampliado usando a técnica de *data augmentation* para o treinamento. O ajuste fino do modelo *Cellpose* foi realizado com 100 imagens por meio do *Cellpose GUI*, que permite treinar modelos customizados de forma interativa, enquanto o classificador foi treinado sobre imagens individuais recortadas a partir dos *bounding boxes* obtidos do *dataset* original. Além disso, desenvolveu-se uma aplicação *web* com *backend* em *Python* e *frontend* em *React*, permitindo ao usuário carregar múltiplas imagens, revisar classificações e obter automaticamente o índice mitótico. Os resultados demonstram que o *pipeline* é capaz de identificar e classificar células com uma eficiência satisfatória, atingindo uma precisão média de 87% na segmentação e acurácia global de 89% na classificação, reduzindo significativamente o tempo necessário para análise citogenética laboratorial. O sistema desenvolvido representa um avanço no apoio à automação de análises microscópicas e estabelece uma base para a construção de um *dataset* próprio e melhorias futuras.

Palavras-chave: Citogenética; *Deep Learning*; *Cellpose*; *EfficientNet*; Automação; Mitose; *Allium cepa*.

ABSTRACT

The cytogenetic analysis of plant cells is a fundamental step in studies of environmental toxicology, genetics, and plant physiology. However, the traditional process of identifying mitotic phases in cytological slides is manual, time-consuming, and prone to inter-observer variability. This work proposes an automated pipeline for the segmentation and classification of cells across different stages of cell division, integrating two deep learning approaches: the Cellpose model for cell segmentation and the EfficientNet architecture for cytological classification. The dataset comprised 656 microscopic images containing cells from *Allium cepa* and *Lactuca sativa*, from which 993 individual cell crops were initially extracted, including 442 interphase cells and the remainder distributed among the mitotic phases. This dataset was subsequently expanded using data augmentation techniques for training. Fine-tuning of the Cellpose model was performed using 100 images through the Cellpose graphical user interface, which enables interactive training of customized models, while the classifier was trained on individual cell images cropped from bounding boxes obtained from the original dataset. In addition, a web application was developed with a Python backend and a React frontend, allowing users to upload multiple images, review classification results, and automatically compute the mitotic index. The results demonstrate that the proposed pipeline is capable of identifying and classifying cells with satisfactory efficiency, achieving an average segmentation precision of 87% and a global classification accuracy of 89%, while significantly reducing the time required for cytogenetic laboratory analysis. The developed system represents an advancement in the automation of microscopic analyses and establishes a foundation for the construction of a dedicated dataset and future improvements.

Keywords: Cytogenetics; Deep Learning; Cellpose; EfficientNet; Automation; Mitosis; *Allium cepa*.

LISTA DE TABELAS

Tabela 1 - Relatório de Classificação do Modelo <i>EfficientNet-B0</i> no conjunto de validação pós <i>fine-tuning</i>	35
--	----

LISTA DE FIGURAS

Figura 1 - Fases do ciclo celular.....	13
Figura 2 - Diagrama de Fluxo do Sistema.....	16
Figura 3 - Exemplos de imagens do dataset base.....	17
Figura 4 - Exemplo de imagens obtidas no BIOGEN.....	17
Figura 5 - Exemplo do processo de segmentação manual através.....	18
da interface GUI Cellpose.....	18
Figura 6 - Exemplos de operações de data augmentation.....	21
Figura 7 - Esquema da Arquitetura - Requisição do Cliente.....	25
Figura 8 - Esquema da Arquitetura - Resposta do Servidor.....	25
Figura 9 - Esquema Macro Backend - Processamento da Requisição.....	26
Figura 10 - Pseudocódigo Algoritmo de Extração de Bounding Box.....	27
Figura 11 - Interface Gráfica.....	29
Figura 12 - Componente de Upload da Imagem.....	30
Figura 13 - Componente de Detecção de Células e Seleção de Fases.....	30
Figura 14 - Componente de Visualização da Imagem,.....	31
Classificações e Ferramentas de Edição.....	31
Figura 15 - Componente de Envio dos Resultados.....	32
Figura 16 - Matriz de Confusão do modelo no conjunto de validação.....	36

LISTA DE GRÁFICOS

Gráfico 1 - Distribuição de frequência das imagens no.....	20
<i>dataset</i> separadas por fase celular.....	20
Gráfico 2 - Distribuição de frequência das imagens de treinamento.....	21
<i>com o data augmentation</i> aplicado.....	21
Gráfico 3 - Evolução de perda e acurácia (Fase 1 e 2).....	34

LISTA DE QUADROS

Quadro 1 - Hiperparâmetros de Treinamento do Modelo Cellpose.....	19
Quadro 2 - Hiperparâmetros de treinamento do modelo EfficientNet-B0.....	22
Quadro 3 - Relação dos requisitos funcionais.....	28

LISTA DE ABREVIATURAS

BIOGEN	Laboratório de Biotecnologia Ambiental e Genotoxicidade
CNN	Rede Neural Convolucional
IM	Índice Mitótico
mAP	Precisão Média
UNIFAL-MG	Universidade Federal de Alfenas

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 OBJETIVO.....	13
2 REVISÃO BIBLIOGRÁFICA.....	14
3 METODOLOGIA.....	16
3.1 PREPARAÇÃO E CONSOLIDAÇÃO DO <i>DATASET</i>	16
3.1.1 CONJUNTO DE IMAGENS.....	17
3.2 PROCESSO DE SEGMENTAÇÃO E GERAÇÃO DE MÁSCARAS.....	18
3.3 <i>DATASET</i> DE CLASSIFICAÇÃO.....	19
3.3.1 CONSTRUÇÃO E CURADORIA DO <i>DATASET</i>.....	19
3.3.2 APLICAÇÃO DE <i>DATA AUGMENTATION</i>.....	20
3.4 TREINAMENTO DA REDE EFFICIENTNET-B0.....	22
3.5 PROCEDIMENTO DE AVALIAÇÃO DO MODELO.....	23
4 DESENVOLVIMENTO.....	24
4.1 APLICAÇÃO WEB.....	24
4.1.1 MODELO DE IMPLEMENTAÇÃO.....	24
4.1.2 ARQUITETURA DA APLICAÇÃO.....	24
4.1.3 <i>BACKEND</i>.....	26
4.1.4 INTERFACE GRÁFICA (<i>FRONTEND</i>).....	27
4.1.5 REQUISITOS FUNCIONAIS.....	28
4.1.6 INTERFACE E COMPONENTES VISUAIS.....	29
4.1.7 HOSPEDAGEM DA APLICAÇÃO WEB.....	32
5 RESULTADOS.....	34
5.1. ANÁLISE DAS MÉTRICAS POR CLASSE.....	35
5.2 INTERPRETAÇÃO DA MATRIZ DE CONFUSÃO.....	36
5.3 ANÁLISE COMPARATIVA COM O ESTADO DA ARTE.....	37
6 CONCLUSÃO E TRABALHOS FUTUROS.....	39
REFERÊNCIAS.....	40

1 INTRODUÇÃO

A análise do índice mitótico constitui uma ferramenta central na citogenética experimental, pois permite avaliar o nível de proliferação celular e identificar alterações decorrentes da ação de agentes citotóxicos e genotóxicos. Em contextos experimentais, variações nesse índice são amplamente empregadas como biomarcadores de citotoxicidade e genotoxicidade, sendo utilizadas na avaliação de substâncias químicas, extratos de plantas medicinais e potenciais carcinógenos (Frota et al., 2019). Dessa forma, o índice mitótico ultrapassa o caráter didático e assume papel fundamental em estudos laboratoriais e ambientais.

Para que essa avaliação seja confiável, recomenda-se a análise de um número expressivo de células. Segundo Nicută et al. (2025), devem ser examinadas entre 1.000 e 5.000 células por lâmina microscópica, tanto em amostras testadas quanto em controles. Esse procedimento possibilita identificar regiões de maior atividade mitótica e detectar alterações significativas no padrão de divisão celular.

Entretanto, a determinação manual do índice mitótico constitui tarefa laboriosa, exigindo profissionais especializados que examinem manualmente centenas ou milhares de células sob microscópio, contabilizando aquelas em divisão versus em interfase, além de estar sujeita à variabilidade intra- e interobservador (Orchid; Puthanpurayil, 2016). Com isso, a classificação visual das fases mitóticas apresenta desafios significativos, mesmo para especialistas. A distinção entre células em interfase e em prófase inicial, por exemplo, é sutil e frequentemente comprometida por variações na preparação das lâminas, o que pode resultar em inconsistências nos resultados (Donovan et al., 2021).

Dentre as estratégias disponíveis, as Redes Neurais Convolucionais (CNNs) têm se destacado. Diferente de métodos tradicionais que exigem que o pesquisador defina manualmente o que o computador deve procurar (como tamanho ou forma), as CNNs aprendem a extrair as características visuais mais relevantes diretamente das imagens, permitindo uma classificação mais robusta dos diferentes fenótipos celulares.

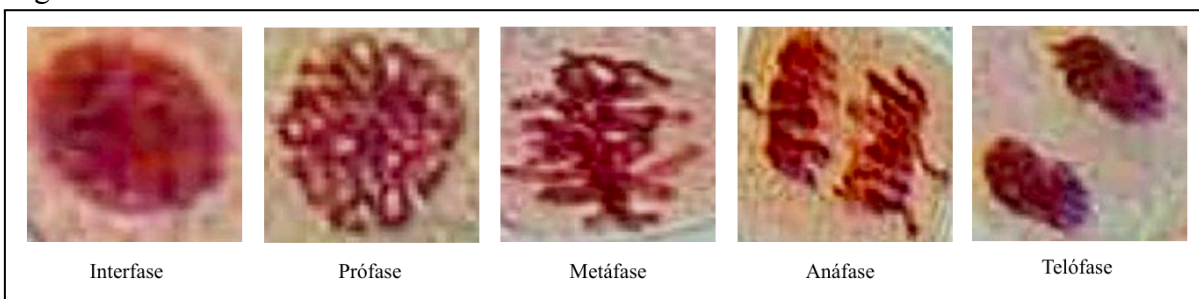
Nesse contexto, a análise de células vegetais constitui a base experimental para o estudo da divisão celular e da citogenética. Espécies como a cebola (*Allium cepa* L.) e a alface (*Lactuca sativa* L.) são amplamente utilizadas como modelos biológicos, uma vez que suas raízes, especialmente a região meristemática apical, oferecem condições ideais para a observação do processo mitótico, consolidando esses organismos como padrões em

laboratórios de pesquisa (Nicută et al., 2025).

O índice mitótico (IM) é definido como a razão entre o número de células em divisão mitótica e o número total de células observadas em determinada área meristemática, sendo geralmente expresso em percentual (Meuten; Moore; George, 2016).

O ciclo celular dessas células é composto pela interfase e pela mitose, a qual se divide em prófase, metáfase, anáfase e telófase. Essas fases apresentam características morfológicas bem definidas e podem ser identificadas por meio de microscopia óptica, utilizando técnicas simples de coloração, como orceína acética ou Giemsa (Brammer et al., 2015). Na Figura 1 é possível observar as fases do ciclo celular bem delimitadas.

Figura 1 - Fases do ciclo celular



Fonte: Autores (2025).

1.1 OBJETIVO

O objetivo principal deste trabalho é otimizar a análise citogenética em ambientes laboratoriais e superar os desafios comuns à rotina de pesquisa global nessa área. O trabalho consiste na implementação das etapas essenciais do processo, como o pré-processamento das imagens, a segmentação das células usando o modelo *Cellpose*, a classificação das fases do ciclo celular por meio do *EfficientNet*, e por fim, a construção de uma aplicação *web* para a integração do *pipeline*. Além de reduzir a dependência da contagem manual, o sistema atua como uma ferramenta complementar aos métodos automatizados já empregados, como citômetros de fluxo, contribuindo para a identificação e correção de inconsistências oriundas de limitações instrumentais ou de aquisição de dados.

O êxito na consolidação desse *pipeline* e na construção de uma ferramenta funcional tem potencial para fortalecer as práticas de análise citogenética. Assim, busca-se iniciar a criação de uma base tecnológica sólida que permita maior agilidade, padronização e confiabilidade nas rotinas de pesquisa, ampliando a capacidade da instituição de inovar e produzir conhecimento na interface entre biotecnologia e computação.

2 REVISÃO BIBLIOGRÁFICA

O estudo de Balagbis et al. (2024) demonstrou o potencial da arquitetura *YOLOv8* para a identificação e classificação de células em divisão e não em divisão, em imagens de meristemas radiculares de *Allium cepa*. O modelo apresentou desempenho expressivo, alcançando precisão média (mAP) de 88,33%, precisão de 85,9% e recall de 83%, evidenciando que técnicas de visão computacional possuem capacidade concreta para automatizar etapas tradicionalmente manuais da análise citogenética. A estratégia de detecção em estágio único característica do *YOLOv8* oferece vantagens relevantes, como elevada velocidade de inferência e integração direta entre identificação e localização das células em um único processamento.

Entretanto, o estudo também evidenciou limitações importantes, especialmente em imagens com alta densidade celular. Os autores observaram perda de instâncias quando muitas células estavam presentes no mesmo campo, resultando em contagens inferiores às reais. Nesses cenários, embora o modelo mantivesse bom desempenho na classificação das células detectadas, não conseguia reconhecer todas as células existentes, indicando um problema de subdetecção.

Em análises citogenéticas, essa limitação é significativa, já que uma subdetecção pode comprometer cálculos importantes como índice mitótico, frequência de anomalias cromossômicas ou proporção de fases mitóticas. Esses desafios reforçam a motivação do presente trabalho em investigar uma abordagem híbrida, na qual a etapa de detecção é aprimorada por meio da segmentação oferecida pelo modelo *Cellpose*, permitindo melhor delimitação das células mesmo em imagens com elevada densidade celular.

Embora arquiteturas híbridas que combinam caminhos de processamento *CNN* e *Transformer*, como o *HyFormer* (Jiang et al., 2024), representem a fronteira contemporânea em análise de imagem biomédica com desempenho superior em segmentação multi-classe, estas abordagens frequentemente exigem volumes massivos de dados e alto custo computacional. O presente trabalho foca na eficiência e robustez de *CNNs* consolidadas, utilizando a *EfficientNet* como referência para um *dataset* de porte médio, estabelecendo uma base sólida e computacionalmente acessível antes de introduzir a complexidade de mecanismos de atenção.

O trabalho de Stringer et al. (2021) apresenta o *Cellpose*, um algoritmo generalizador baseado em aprendizado profundo para segmentação celular que foi treinado em um conjunto

de dados heterogêneo contendo mais de 70.000 objetos segmentados obtidos de diversas modalidades microscópicas. Este método destaca-se por sua capacidade de segmentar células de uma ampla variedade de tipos de imagens sem necessidade de retreinamento ou ajustes de parâmetros, utilizando uma arquitetura *U-Net* modificada que prediz campos de fluxo horizontal e vertical juntamente com máscaras binárias de células.

Com isso, o modelo *Cellpose* demonstra excelentes resultados em segmentação de estruturas celulares, particularmente em imagens de fluorescência com marcação citoplasmática ou de membrana. Este trabalho incorporou esta metodologia como componente da estratégia de segmentação, reconhecendo sua robustez e aplicabilidade em diferentes contextos de imagem.

A classificação celular exige a análise de padrões visuais complexos. Enquanto a literatura clássica (Bourn et al., 2025) destaca métricas geométricas como área e perímetro para fins classificatórios, a distinção entre fases da mitose (como prófase inicial e interfase) depende de texturas sutis da cromatina que a morfologia simples pode não capturar.

A *EfficientNet*, proposta por Tan e Le (2019), utiliza um método de escalonamento composto que escala simultaneamente a profundidade, a largura e a resolução da rede, permitindo alcançar alta precisão com eficiência computacional. Este trabalho adota essa arquitetura como *backbone* para extração de características e classificação, aproveitando sua capacidade de aprender representações discriminativas de imagens em múltiplas escalas, o que se revela particularmente vantajoso para dados de dimensionalidade limitada ou com variabilidade morfológica reduzida.

3 METODOLOGIA

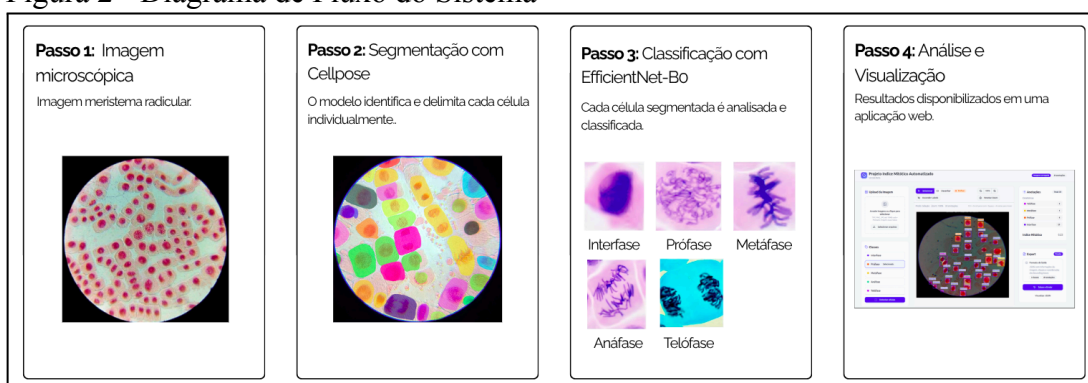
Este trabalho, focado na automação da análise citogenética de células vegetais, adotou uma abordagem metodológica sistemática, composta por fases distintas, para estruturar os três eixos centrais da pesquisa: consolidação de dados, modelagem de inteligência artificial e aplicação prática. O processo de desenvolvimento foi estruturado em três etapas sequenciais:

- a) preparação das imagens do *dataset*;
- b) desenvolvimento e treinamento da arquitetura híbrida;
- c) implementação de uma aplicação *web*.

Nesse contexto, a Figura 2 ilustra de forma esquemática o fluxo do *pipeline* construído, evidenciando as etapas que operacionalizam os objetivos deste trabalho. O processo tem início com a aquisição da imagem microscópica do meristema radicular, que constitui a base para as análises subsequentes. Em seguida, aplica-se o modelo para a segmentação das células, permitindo a identificação individual de cada estrutura celular presente na lâmina.

As células segmentadas são então submetidas à etapa de classificação responsável por atribuir a cada célula uma das fases do ciclo celular. Por fim, os resultados são consolidados e disponibilizados em uma aplicação web, possibilitando a visualização interativa, a auditoria humana das predições e o cálculo automatizado do índice mitótico.

Figura 2 - Diagrama de Fluxo do Sistema



Fonte: Autores (2025).

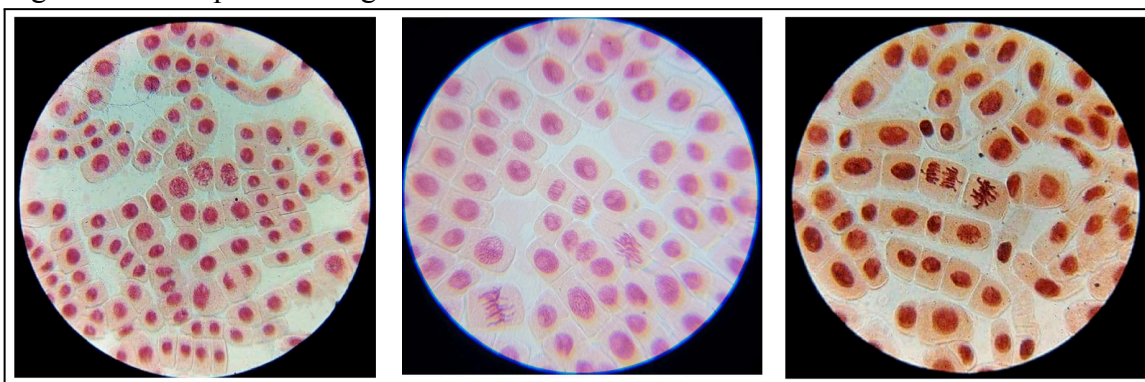
3.1 PREPARAÇÃO E CONSOLIDAÇÃO DO *DATASET*

O desenvolvimento do sistema baseou-se em uma estratégia de dados em duas etapas, correspondendo aos módulos de segmentação e classificação do *pipeline*.

3.1.1 CONJUNTO DE IMAGENS

Para o treinamento e ajuste fino dos modelos, foi consolidado um conjunto de imagens microscópicas de lâminas inteiras de *Allium cepa* e de *Lactuca sativa*. A base principal foi obtida do *dataset* público disponibilizado por Balagbis et al. (2024) na plataforma *Roboflow*, composto por 640 imagens de *Allium cepa*. Para facilitar a organização do texto, este conjunto de dados será chamado, a partir deste ponto, de “*dataset* base”. A Figura 3 apresenta alguns exemplos representativos desse conjunto.

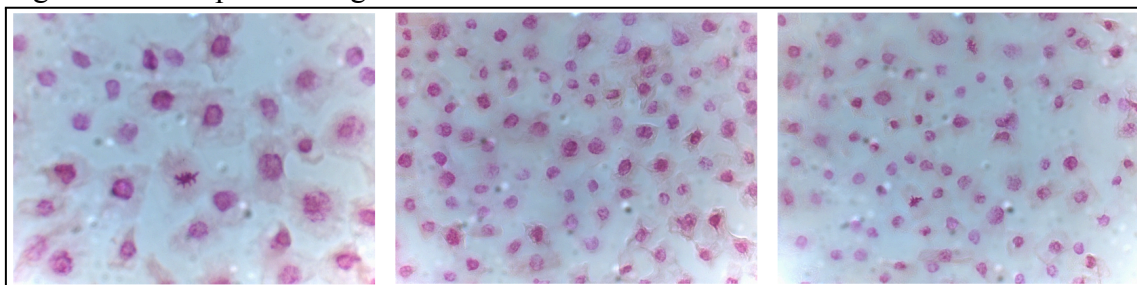
Figura 3 - Exemplos de imagens do *dataset* base



Fonte: Autores (2025).

A este *dataset* base, foram incorporadas 16 imagens de *Lactuca sativa* obtidas no BIOGEN. A Figura 4 apresenta exemplos destas amostras, que possuem características específicas de captura do laboratório local.

Figura 4 - Exemplo de imagens obtidas no BIOGEN



Fonte: Autores (2025).

Embora reduzido em número, este subconjunto foi essencial para calibrar o modelo às condições específicas de captura e iluminação e coloração do laboratório local. O conjunto total foi submetido a uma validação visual por especialistas para remoção de artefatos e segmentações ruidosas, consolidando o *dataset* final com um total de 656 imagens.

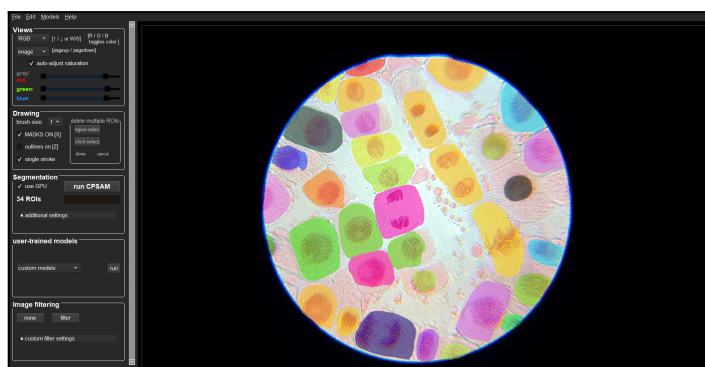
3.2 PROCESSO DE SEGMENTAÇÃO E GERAÇÃO DE MÁSCARAS

O treinamento do modelo de segmentação foi realizado utilizando a interface gráfica nativa do *Cellpose*, aproveitando sua funcionalidade de *human-in-the-loop* para refinamento interativo. Para esta etapa, foram utilizadas 100 imagens do *dataset* como dados de teste/validação, diretamente pela interface.

A Figura 5 apresenta a interface visual do *Cellpose* empregada no processo de segmentação e correção manual. Após a segmentação automática inicial, cada célula individualizada é destacada por uma máscara, que consiste em uma representação digital da região ocupada pela célula na imagem. Em termos práticos, a máscara funciona como um “contorno” ou “marcação” que delimita, pixel a pixel, a área correspondente a cada célula, separando-a do fundo e das células vizinhas.

Essas máscaras são exibidas em cores distintas, facilitando a inspeção visual e permitindo ao usuário identificar erros de segmentação, como sobreposição ou fragmentação indevida de células. A partir dessa visualização, o pesquisador pode corrigir manualmente as delimitações, ajustando as máscaras diretamente sobre a imagem por meio das ferramentas fornecidas pelo software, contribuindo para o aprimoramento progressivo do modelo de segmentação.

Figura 5 - Exemplo do processo de segmentação manual através da interface *GUI Cellpose*



Fonte: Autores (2025).

Os parâmetros de treinamento configurados na interface, conforme representado no Quadro 1, foram:

Quadro 1 - Hiperparâmetros de Treinamento do Modelo *Cellpose*.

Hiperparâmetro	Configuração
Modelo Inicial	<i>cyto 2</i>
Canal Principal	2 (Verde)
Canal Secundário	1 (Vermelho)
Decaimento de Peso	0.0001
Taxa de Aprendizado	0,1
Épocas	100

Fonte: Autores (2025).

3.3 DATASET DE CLASSIFICAÇÃO

3.3.1 CONSTRUÇÃO E CURADORIA DO DATASET

Para a composição do conjunto de imagens destinado à etapa de classificação, realizou-se o recorte individual das células. No *dataset* base, esse processo utilizou as coordenadas dos arquivos de anotação no padrão *YOLOv8* disponibilizados juntamente ao *dataset*. Por sua vez, para as imagens provenientes do BIOGEN, o recorte baseou-se nas máscaras geradas pela segmentação prévia com o *Cellpose*.

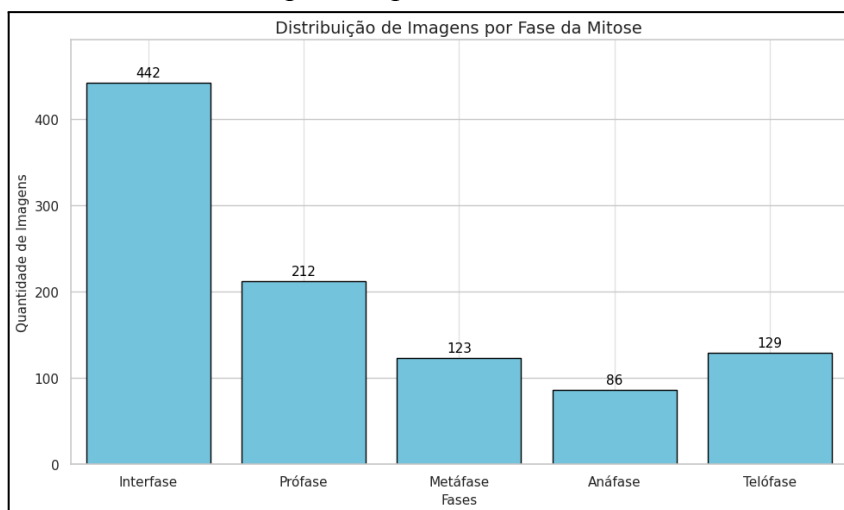
Visando viabilizar uma análise citogenética aprofundada, foi realizada uma reclassificação manual e especializada de todo o acervo. Especialistas do BIOGEN revisaram as imagens do *dataset* base, convertendo a classificação binária original para o sistema biológico de cinco fases. As células originalmente rotuladas como ND (Não Divisão) foram mapeadas para a classe interfase, enquanto as células marcadas como D (Divisão) foram discriminadas em suas fases específicas: prófase, metáfase, anáfase e telófase.

O esforço de curadoria resultou na consolidação de um *dataset* final composto por 993 imagens recortadas de células. A estratificação das fontes de dados revelou que o *dataset* base contribuiu com a maior parcela do acervo, totalizando 896 imagens de recortes, das quais 715 foram destinadas ao conjunto de treinamento e 181 ao de teste. De forma complementar, a fonte do BIOGEN adicionou 97 imagens de recortes, divididos em 68 imagens para a etapa de treinamento e 29 imagens para a fase de teste.

Ressalta-se que, embora o *dataset* tenha incorporado amostras de *Lactuca sativa*, provenientes do acervo do BIOGEN com o objetivo de ampliar a variabilidade morfológica e a robustez do classificador frente a diferentes condições de captura, o foco principal da validação e comparação com o estado da arte recai sobre a espécie *Allium cepa*. A segregação detalhada da performance do modelo por espécie vegetal não foi objeto deste estudo,

constituindo uma oportunidade relevante para investigações futuras. A distribuição final das classes neste conjunto pode ser visualizada no Gráfico 1.

Gráfico 1 - Distribuição de frequência das imagens no *dataset* separadas por fase celular



Fonte: Autores (2025).

3.3.2 APLICAÇÃO DE *DATA AUGMENTATION*

Para ampliar a variabilidade das imagens presentes no conjunto de treinamento e, consequentemente, reduzir o risco de *overfitting*, foi aplicado um processo de *data augmentation* exclusivamente nesta partição, utilizando a classe *ImageDataGenerator* da biblioteca *Python TensorFlow/Keras*¹.

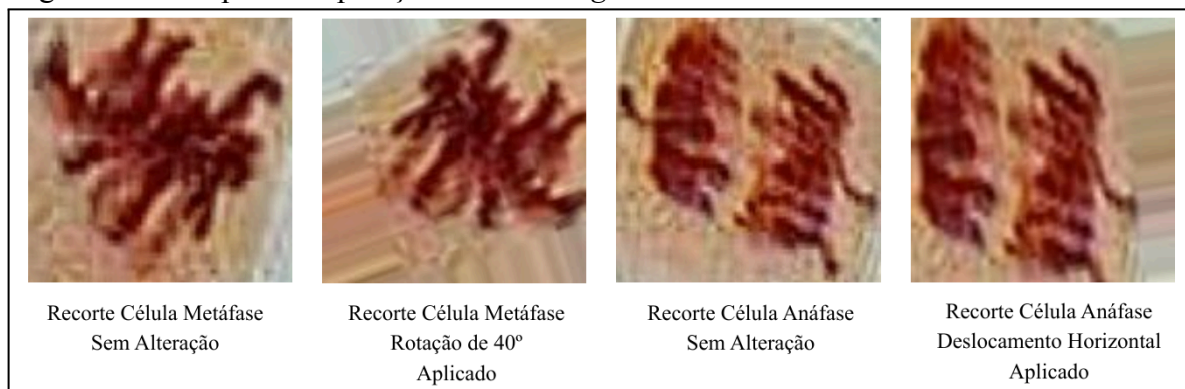
Inicialmente, definiu-se um diretório contendo as imagens originais do subconjunto de treino e um diretório de saída destinado ao armazenamento das imagens aumentadas. O procedimento gerou dez novas variações para cada imagem de treino, aplicando um fator de aumento de 10x, incorporando assim transformações aleatórias que simulam condições diversas de captura.

As operações de aumento incluíram rotações de até 40°, deslocamentos horizontais e verticais de 20%, cisalhamento, *zoom* em $\pm 20\%$, além de inversão horizontal. Para o tratamento das áreas vazias resultantes das transformações geométricas, adotou-se o método de preenchimento pelo vizinho mais próximo, visando preservar a consistência das características visuais das células.

¹ <https://www.tensorflow.org/guide/keras?hl=pt-br>

A Figura 6 exemplifica as operações de aumento de dados aplicadas ao conjunto de imagens. À esquerda, visualiza-se uma célula em metáfase em seu estado original, seguida pela mesma imagem submetida a uma rotação de 40°. Na sequência, é apresentada uma célula em anáfase sem alterações, comparada, na imagem adjacente, ao resultado da aplicação de uma operação de deslocamento horizontal.

Figura 6 - Exemplos de operações de *data augmentation*

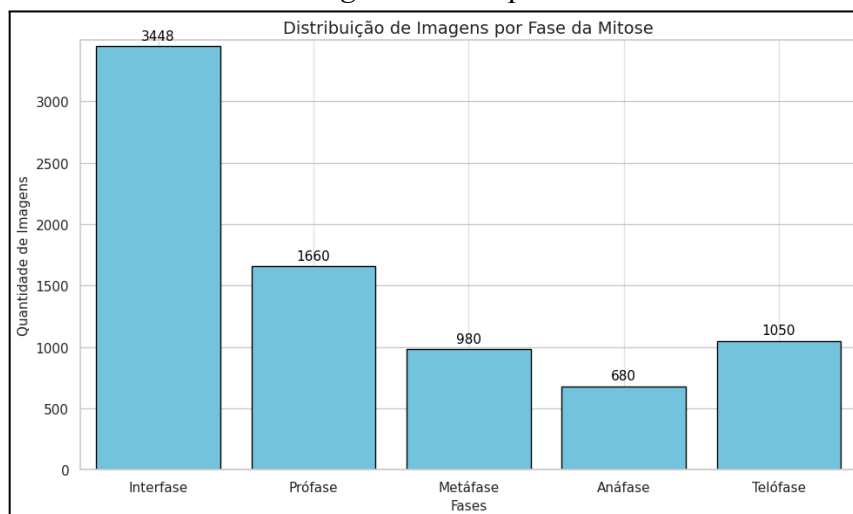


Fonte: Autores (2025).

A incorporação dessas modificações ampliou a diversidade visual do *dataset* de treinamento, permitindo que o classificador se tornasse mais resistente às variações naturais de posicionamento, iluminação e orientação, enquanto o conjunto de teste foi mantido com as imagens originais para garantir uma avaliação fidedigna do desempenho do modelo.

O Gráfico 2 apresenta a distribuição de frequência das imagens de treinamento após a aplicação do *data augmentation*. É possível observar que houve um aumento considerável no número total de imagens.

Gráfico 2 - Distribuição de frequência das imagens de treinamento com o *data augmentation* aplicado



Fonte: Autores (2025).

3.4 TREINAMENTO DA REDE *EFFICIENTNET-B0*

Após a aplicação do *data augmentation*, o volume de dados resultou em um total de 7.818 imagens de treino. O conjunto de teste permaneceu com as imagens originais, totalizando 210 amostras, garantindo que a validação ocorresse sobre dados reais e não sintéticos.

O treinamento foi conduzido utilizando a técnica de *transfer learning* em dois estágios (*Feature Extraction* seguido de *Fine-Tuning*), com a arquitetura *EfficientNet-B0* pré-treinada na *ImageNet*. Para mitigar o desbalanceamento natural entre as fases da mitose, com predominância da interfase, utilizou-se a estratégia de pesos de classe (*Class Weights*) na função de perda, penalizando com maior severidade os erros nas classes minoritárias, como anáfase, metáfase e telófase.

Os hiperparâmetros e as configurações estruturais adotadas para o treinamento da rede neural foram detalhadas no Quadro 2. O modelo foi compilado com o otimizador Adam e a função de perda *Cross Entropy Loss*, a qual foi adaptada com a técnica de *Class Weights*.

Quadro 2 - Hiperparâmetros de treinamento do modelo *EfficientNet-B0*

Hiperparâmetro	Configuração
Arquitetura Base	<i>EfficientNet-B0</i> (Pré-treinada - pesos <i>ImageNet</i>)
Estratégia	<i>Transfer Learning</i> e <i>Fine-Tuning</i>
Otimizador	<i>Adam</i>
Função de Perda	<i>Cross Entropy Loss</i>
Taxa de Aprendizado	0,001 e 0,0001 no <i>Fine-Tuning</i>

Regularização	<i>Early Stopping</i> (Paciência de 3 a 7 épocas)
Tamanho do Lote	32

Fonte: Autores (2025).

Inicialmente, a taxa de aprendizado foi fixada em 0,001 para o treinamento do classificador linear, mantendo-se as camadas da base congeladas para preservar as características genéricas. Em um segundo momento, procedeu-se ao ajuste fino, descongelando-se as camadas e reduzindo a taxa de aprendizado para 0,0001, permitindo a adaptação dos pesos às especificidades das células vegetais.

Todo o processo foi executado com um tamanho de lote de 32 amostras. Diferente de uma abordagem com número fixo de iterações, o treinamento foi controlado pelo mecanismo de *Early Stopping*, configurado para interromper a execução caso a perda no conjunto de validação não apresentasse decaimento após um período de paciência (definido entre 3 e 7 épocas), garantindo a seleção do modelo com maior capacidade de generalização e evitando o *overfitting*.

3.5 PROCEDIMENTO DE AVALIAÇÃO DO MODELO

A avaliação do desempenho do *pipeline* construído foi conduzida de forma sistemática, utilizando um conjunto de teste independente composto por 210 imagens não vistas durante o treinamento. Inicialmente, as imagens foram submetidas ao modelo de segmentação Cellpose ajustado, responsável pela geração automática das máscaras celulares.

Em seguida, as regiões segmentadas foram extraídas e encaminhadas ao classificador baseado na arquitetura *EfficientNet-B0*, previamente treinado em duas fases: uma etapa inicial com camadas congeladas e uma etapa posterior de fine-tuning com liberação total dos pesos da rede.

As predições do classificador foram comparadas com os rótulos de referência anotados manualmente, permitindo a construção da matriz de confusão multiclasse. A partir dessa matriz, foram calculadas as métricas de Precisão, Sensibilidade (*Recall*), Especificidade e F1-Score para cada classe mitótica, além da acurácia global do modelo.

Adicionalmente, a evolução do aprendizado ao longo das épocas foi analisada por meio das curvas de perda e acurácia, considerando separadamente as duas fases de treinamento, com o objetivo de avaliar o impacto do processo de ajuste fino no desempenho final do modelo.

4 DESENVOLVIMENTO

4.1 APLICAÇÃO *WEB*

Nesta subseção, apresentamos a construção da aplicação *web* utilizada para integrar o *pipeline*. Nele, descrevemos o modelo, a arquitetura do sistema, o desenvolvimento do *backend* e a interface gráfica (*frontend*). Ao final, discutimos a hospedagem da aplicação.

4.1.1 MODELO DE IMPLEMENTAÇÃO

Durante a etapa de planejamento, foram analisadas duas arquiteturas possíveis: *software desktop* e modelo cliente-servidor (*web*). A escolha recaiu sobre a segunda opção, priorizando a eficiência no processamento exigido pelo sistema, especificamente quanto ao uso de GPU (*Graphics Processing Unit*).

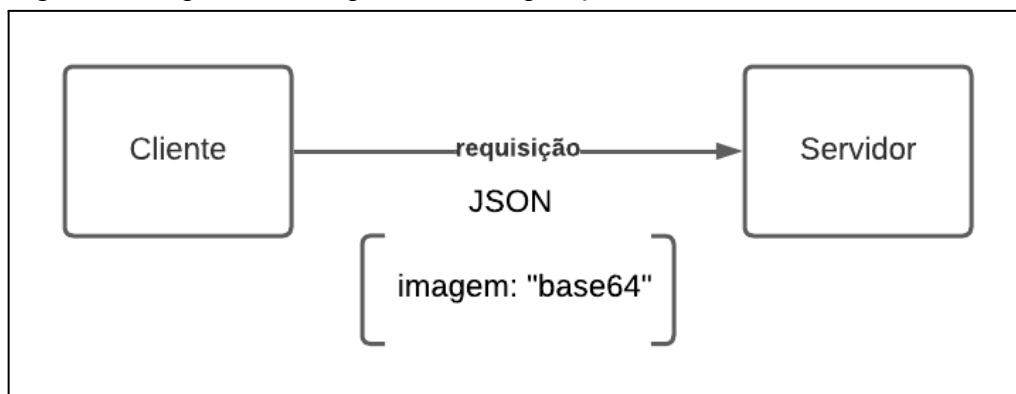
Originalmente voltadas para renderização gráfica e jogos, as GPUs tornaram-se fundamentais na área de inteligência artificial por sua capacidade de realizar cálculos matriciais e operações paralelas, requisitos básicos para redes neurais. Como detalhado anteriormente, o modelo *Cellpose*, adotado neste trabalho, beneficia-se dessa aceleração. A execução via CPU, embora viável, resulta em um processamento significativamente mais lento, inviabilizando o uso prático.

A adoção do modelo *web* democratiza o acesso à ferramenta. Caso optássemos por uma versão *desktop*, isso transferiria ao usuário a obrigatoriedade de possuir um hardware equipado com GPU. No modelo cliente-servidor, o processamento pesado ocorre remotamente, permitindo que o usuário utilize a aplicação via navegador em dispositivos comuns. Adicionalmente, esse formato simplifica a manutenção, pois as atualizações centralizadas no servidor eliminam a necessidade de intervenções técnicas nas máquinas dos usuários.

4.1.2 ARQUITETURA DA APLICAÇÃO

A comunicação entre cliente e servidor adota o protocolo HTTPS, utilizando o formato JSON para a troca de dados, no caso das imagens, aplica-se a codificação em base 64 para o transporte dentro da estrutura de chave-valor. A Figura 7 detalha um exemplo visual dessa interação entre as partes da aplicação.

Figura 7 - Esquema da Arquitetura - Requisição do Cliente

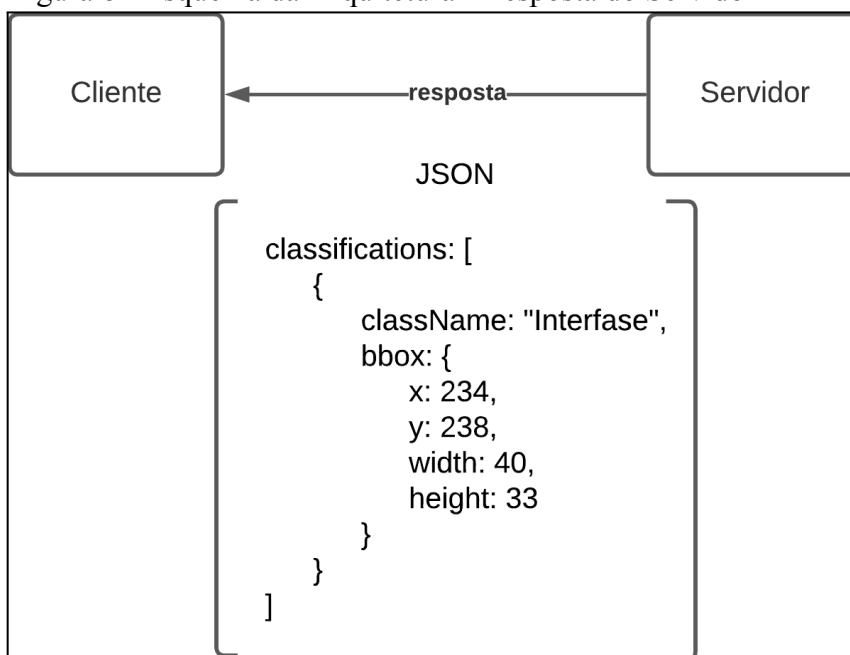


Fonte: Autores (2025).

O retorno do servidor é enviado com o *payload* em formato JSON estruturado a partir do atributo “*classifications*”. Trata-se de uma lista que armazena as detecções realizadas, onde cada objeto é composto por dois atributos: “*className*” (rótulo da classe) e “*bbox*” (geometria da detecção).

O atributo “*bbox*” subdivide-se nas propriedades “*x*” e “*y*”, que indicam a posição do pixel inicial, e “*width*” e “*height*”, que definem a largura e altura da caixa delimitadora. A Figura 8 ilustra a organização hierárquica desses dados.

Figura 8 - Esquema da Arquitetura - Resposta do Servidor



Fonte: Autores (2025).

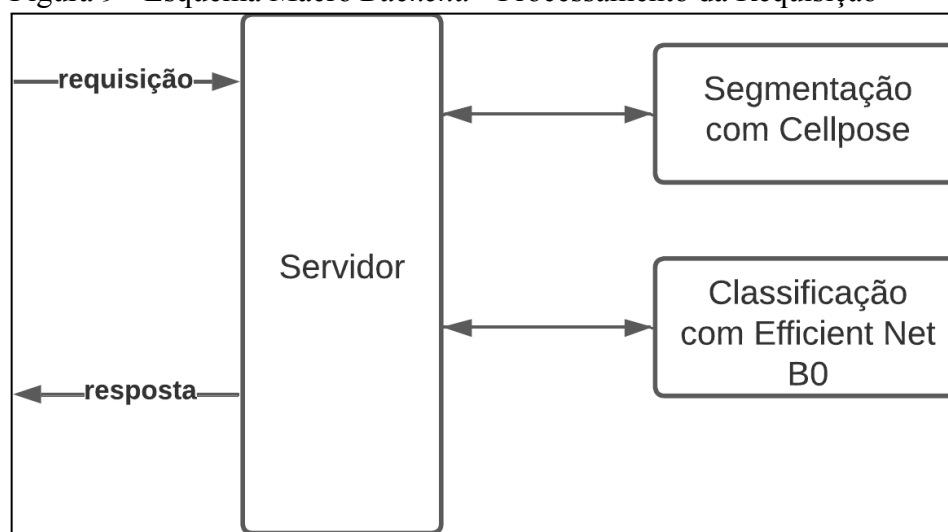
4.1.3 BACKEND

O *backend* da aplicação foi desenvolvido na linguagem *Python*², utilizando o *framework Gradio*³. A escolha por essa tecnologia deve-se ao amplo suporte do *Python* às bibliotecas de redes neurais, incluindo a *EfficientNet*, utilizada neste trabalho. O *Gradio*, por sua vez, foi selecionado por ser compatível com a plataforma de hospedagem adotada, que será discutida em subseção posterior.

A implementação do *backend* desempenha um papel central no sistema, pois é responsável pela integração do *pipeline* de segmentação e classificação de imagens proposto.

A Figura 9 ilustra o esquema macro do *backend*. Ao receber a requisição contendo a imagem, o sistema realiza a segmentação utilizando o modelo Cellpose. A partir das máscaras geradas, é aplicada uma operação de extração dos bounding boxes. Esse processo permite recortar a região de interesse e preparar os dados de entrada (input) para a etapa de classificação.

Figura 9 - Esquema Macro *Backend* - Processamento da Requisição



Fonte: Autores (2025).

A Figura 10 apresenta o pseudocódigo do algoritmo utilizado para o processamento e extração de características das células segmentadas.

² <https://www.python.org/>

³ <https://www.gradio.app/>

Figura 10 - Pseudocódigo Algoritmo de Extração de *Bounding Box*

```

1  PARA cada célula na máscara FACA
2    obter coordenadas dos pixels da célula
3    SE célula vazia ENTAO
4      CONTINUAR
5    FIM SE
6    definir bounding box usando min e max de x e y
7    aplicar padding aos limites (sem ultrapassar bordas
      da imagem)
8    calcular centro da bounding box
9    extrair contorno da célula a partir da máscara binária
10   simplificar contorno e converter para lista de
      pontos
11   RETORNAR: bounding box, contorno, centro
12   FIM PARA

```

Fonte: Autores (2025).

O fluxo de execução consiste em iterar sobre cada célula identificada na máscara; caso a célula não possua dados válidos (esteja vazia), a iteração é ignorada e o ciclo avança para o próximo elemento.

Para as células válidas, o algoritmo define a região de interesse calculando uma *bounding box* com base nas coordenadas extremas, aplicando um *padding* (margem) de 15 pixels. Essa margem garante que a célula esteja inteiramente contida no recorte, respeitando as bordas da imagem. Na sequência, são calculados o centro geométrico e o contorno a partir da máscara binária, sendo este último simplificado em uma lista de vetores. Ao final, o algoritmo retorna o *bounding box*, o contorno e o centro para as etapas subsequentes.

Com as caixas delimitadoras definidas, cada célula é recortada, padronizada e submetida ao classificador, que retorna a fase mitótica correspondente. Por fim, o *backend* consolida todas as informações incluindo as coordenadas originais e as classes preditas em um objeto JSON, devolvendo-o ao frontend.

4.1.4 INTERFACE GRÁFICA (*FRONTEND*)

O *frontend* foi desenvolvido utilizando a linguagem *TypeScript*⁴ e a biblioteca *React*⁵. A escolha por essas tecnologias justifica-se pelo amplo suporte da comunidade e pela agilidade que proporcionam ao desenvolvimento. O *React*, em particular, oferece

⁴ <https://www.typescriptlang.org/>

⁵ <https://react.dev/>

componentes visuais pré-configurados e facilita a organização do código, garantindo a robustez necessária para construir interfaces gráficas complexas e interativas.

Um exemplo dessa complexidade é a seção principal da interface, que abriga o canvas onde serão visualizadas as classificações das células.

4.1.5 REQUISITOS FUNCIONAIS

Para garantir uma interface intuitiva, ou seja, de fácil utilização e entendimento pelo usuário, a aplicação foi projetada considerando a rotina de tarefas de um pesquisador que utilizaria a plataforma. Com base nesse perfil, definiram-se os critérios listados no Quadro 3.

Quadro 3 - Relação dos requisitos funcionais

I - Upload de Imagem	O usuário deve ser capaz de fazer o upload de uma ou mais imagens.
II - Classificação Automática	Ao fazer o upload da imagem o usuário deve ser capaz de solicitar a detecção automática das células na imagem.
III - Índice Mitótico	Após a classificação da células, o índice mitótico deve ser calculado e estar disponível para visualização na interface
IV - Alteração Manual	A interface deve conter ferramentas para que o usuário possa realizar a alteração manual da segmentação e classificação das células em caso de erros na detecção automática.

Fonte: Autores (2025).

O fluxo de uso inicia-se com a funcionalidade de *upload*, que permite a entrada das imagens para análise. Após essa etapa, mediante comando do usuário, o sistema executa a classificação automática, identificando e categorizando as células, processo que serve de base para o cálculo e visualização do índice mitótico.

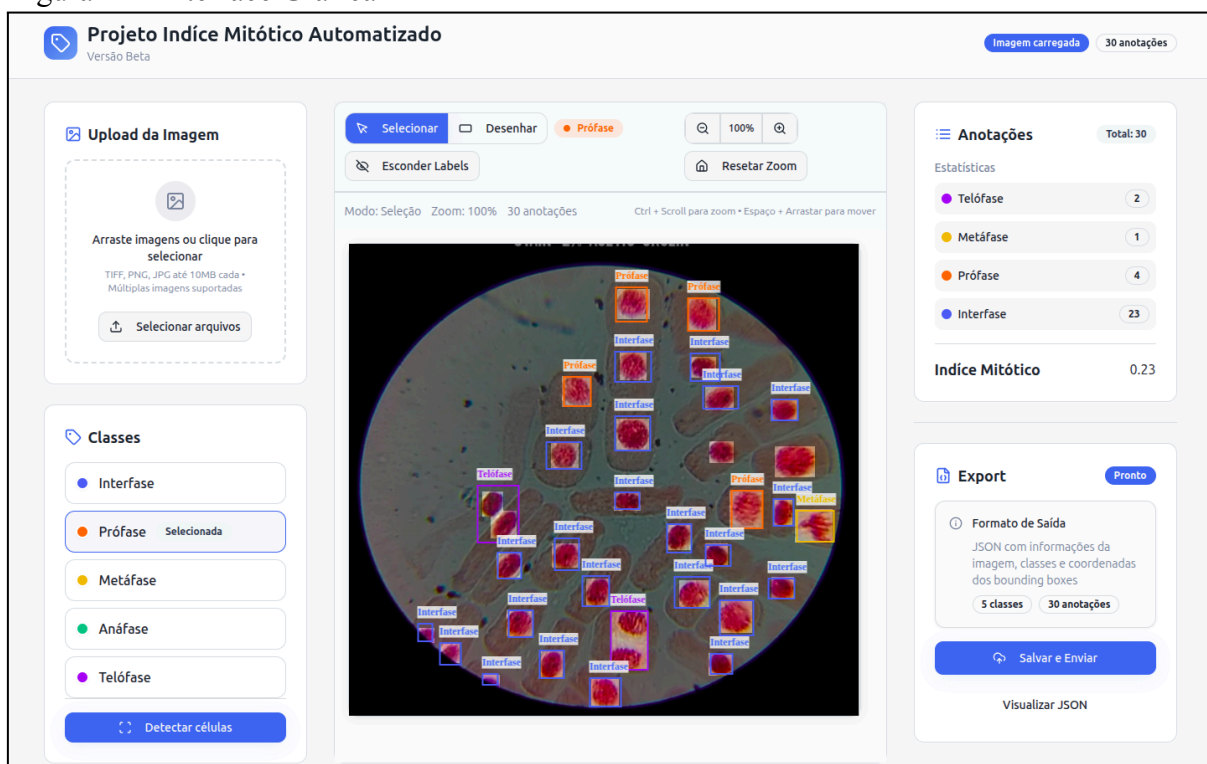
Por fim, para garantir a confiabilidade dos resultados, estabeleceu-se o requisito de alteração manual. Essa funcionalidade oferece ferramentas para que o pesquisador possa intervir e corrigir eventuais inconsistências da detecção automática, permitindo tanto a reclassificação das fases celulares quanto a edição dos *bounding boxes* de segmentação, o que inclui redimensionar, excluir ou adicionar novas delimitações.

4.1.6 INTERFACE E COMPONENTES VISUAIS

A partir dos requisitos estabelecidos, os componentes visuais da aplicação foram implementados. Nesta subseção, descrevemos a interface resultante e seus principais elementos.

A Figura 11 exibe o layout da aplicação após a execução completa do fluxo de trabalho (upload da imagem e detecção automática). Neste estado, observa-se no canto superior direito a contagem de cada fase celular identificada e, imediatamente abaixo, o valor do índice mitótico calculado. A seguir, detalharemos cada componente da interface.

Figura 11 - Interface Gráfica



Fonte: Autores (2025).

A Figura 12 exibe o componente de upload de imagem. Neste campo, o usuário pode selecionar arquivos ou arrastá-los diretamente para a interface. O sistema suporta o envio de múltiplas imagens simultaneamente, nos formatos TIFF, PNG ou JPG, com limite de tamanho de até 10 MB.

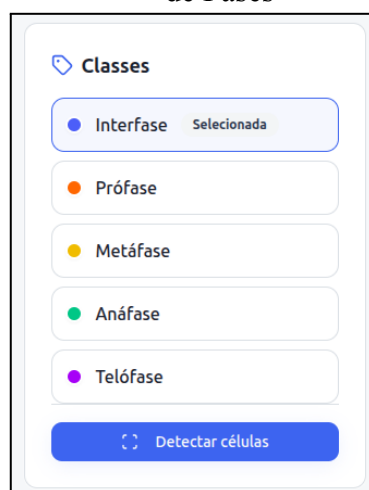
Figura 12 - Componente de
Upload da Imagem



Fonte: Autores (2025).

A Figura 13 ilustra as ferramentas disponíveis para a seleção das fases celulares, permitindo a revisão e o ajuste manual das classificações geradas. Além disso, a interface apresenta o botão responsável por iniciar o processo de detecção automática na imagem.

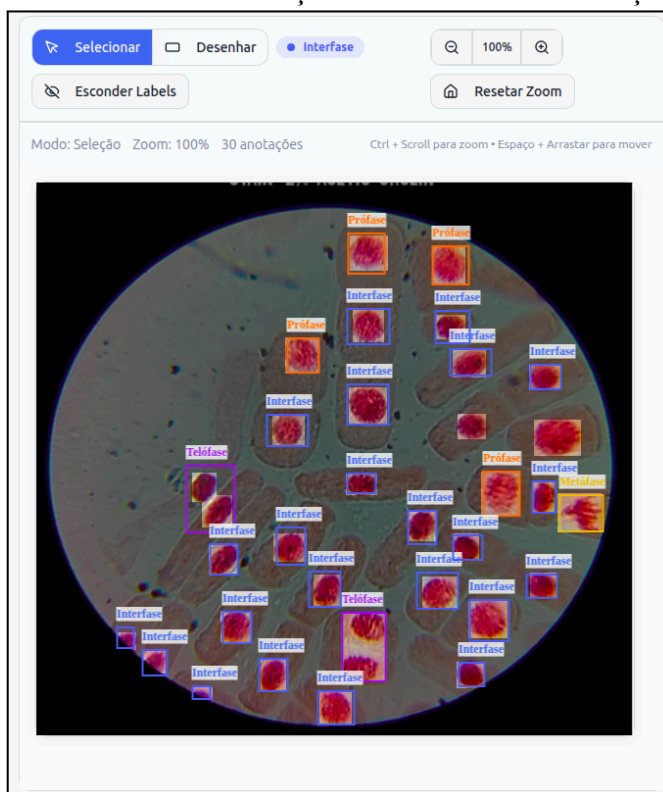
Figura 13 - Componente de Detecção
de Células e Seleção
de Fases



Fonte: Autores (2025).

A Figura 14 apresenta o componente principal da aplicação, responsável pela visualização da imagem e das classificações geradas.

Figura 14 - Componente de Visualização da Imagem, Classificações e Ferramentas de Edição



Fonte: Autores (2025).

Observa-se que as detecções são exibidas no formato de *bounding boxes* ao redor de cada célula. Para facilitar a compreensão, o nome da fase identificada é exibido na parte superior de cada caixa, sendo que cada classe possui uma cor distinta.

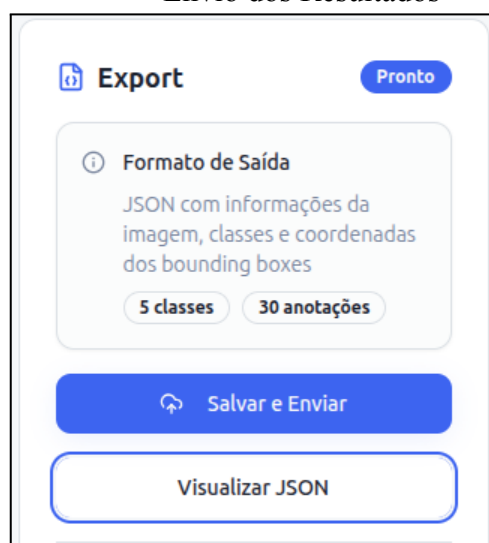
Na parte superior do componente, estão localizadas as ferramentas para edição manual. Com o botão *Selecionar*, o usuário pode escolher uma classificação específica e alterar sua classe (utilizando o componente descrito na Figura 13) ou redimensionar a caixa delimitadora para melhor ajuste à célula.

O botão *Desenhar* permite adicionar novas *bounding boxes* manualmente, contemplando células não identificadas pelo sistema. Paralelamente, o botão *Esconder Labels* desabilita a exibição dos nomes das fases, facilitando a visualização das células em casos de alta densidade.

No canto superior direito, encontram-se os controles de navegação. Os botões de Lupa realizam as funções de *Zoom In* e *Zoom Out* (aproximação e afastamento). Logo abaixo, o botão *Resetar Zoom* restaura a visualização original, recurso útil caso o usuário perca o referencial da imagem durante a manipulação.

Por fim, visando à obtenção de novos dados para o aprimoramento da ferramenta, desenvolveu-se o componente de Envio de Resultados, demonstrado na Figura 15. Após a validação da análise pelo usuário, é possível submeter os dados finais para armazenamento em um bucket S3 na nuvem. Esse processo envia tanto a imagem processada quanto as delimitações e classes em formato JSON, criando um repositório para análises futuras e possíveis melhorias no algoritmo da aplicação.

Figura 15 - Componente de Envio dos Resultados



Fonte: Autores (2025).

4.1.7 HOSPEDAGEM DA APLICAÇÃO WEB

Para tornar o sistema acessível e facilitar sua utilização, a aplicação foi hospedada em serviços de nuvem gratuitos. A implantação separou o *backend* do *frontend* para garantir flexibilidade.

O *backend* foi hospedado no *Hugging Face Spaces*⁶. O motivo dessa escolha é a oferta de suporte à execução com GPU no plano gratuito, essencial para nosso *pipeline*. Contudo, existem limitações: a plataforma impõe um tempo limite diário de uso. Assim, quando o limite é atingido, não é possível processar utilizando a GPU naquele dia. Porém, para a fase inicial e experimental deste projeto, essa cota é suficiente.

O *frontend* foi implantado na *Vercel*⁷, que suporta hospedagem de projetos *React*. Com a integração ao GitHub, é possível subir as alterações rapidamente através do suporte a *CI/CD*

⁶ <https://huggingface.co/spaces>

⁷ <https://vercel.com/>

(*Continuous Integration / Continuous Deployment*) oferecido pela plataforma.

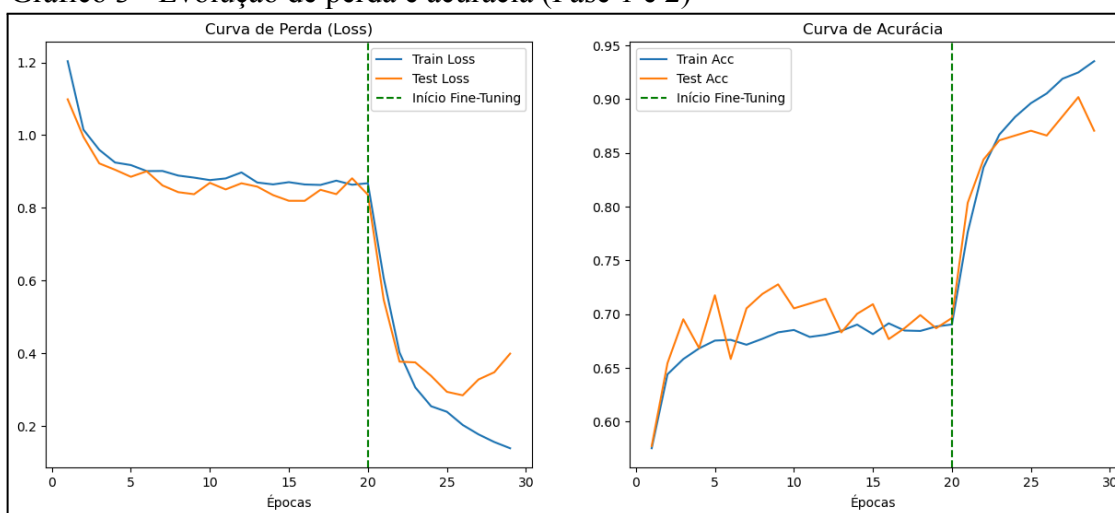
A combinação dessas duas soluções possibilita um sistema funcional, gratuito e de fácil manutenção. Isso permite atualizações constantes, beneficiando os usuários do *pipeline*.

5 RESULTADOS

A implementação do *pipeline* híbrido proposto apresentou resultados quantitativos que validam a viabilidade da automação para a análise citogenética. O processo de ajuste fino aplicado ao modelo *Cellpose* permitiu a segmentação adequada das células, mesmo em regiões de maior densidade, gerando máscaras com aderência suficiente às fronteiras nucleares para a etapa de extração. Após 100 épocas, o modelo atingiu uma precisão média de 87% na detecção de contornos celulares. Este nível de assertividade foi considerado satisfatório para o objetivo do *pipeline*, garantindo a geração automática de máscaras de alta qualidade para a etapa subsequente de classificação.

A avaliação do modelo de classificação no conjunto de teste contendo 210 imagens, demonstrou a eficácia da abordagem de ajuste fino. O classificador atingiu uma acurácia global de 89%, um avanço significativo em relação aos treinamentos basilares. A evolução do aprendizado, considerando as duas fases do treinamento, pode ser observada no Gráfico 3. Nota-se uma queda acentuada na perda e um salto na acurácia a partir da época 20, momento em que as camadas da rede foram descongeladas para o *fine-tuning*, permitindo que o modelo aprendesse características específicas da citologia vegetal.

Gráfico 3 - Evolução de perda e acurácia (Fase 1 e 2)



Fonte: Autores (2025).

5.1. ANÁLISE DAS MÉTRICAS POR CLASSE

Para uma compreensão aprofundada do comportamento do modelo frente à complexidade biológica de cada fase mitótica, as métricas de Precisão, Sensibilidade, Especificidade e F1-Score foram calculadas individualmente. Os dados detalhados encontram-se na Tabela 1.

Tabela 1 - Relatório de Classificação do Modelo *EfficientNet-B0* no conjunto de validação pós *fine-tuning*.

Classe	Precisão	Sensibilidade	Especificidade	F1-Score
Interfase	94,6%	91,7%	95,5%	0,93
Prófase	85,7%	89,4%	95,6%	0,88
Metáfase	90,9%	80,0%	98,9%	0,85
Anáfase	83,3%	83,3%	98,4%	0,83
Telófase	80,0%	95,2%	97,3%	0,87

Fonte: Autores (2025).

A análise da Tabela 1 evidencia um desempenho consistente, correlacionado com as características morfológicas de cada fase. Observou-se que todas as classes apresentaram uma Especificidade superior a 95%, o que demonstra a alta capacidade do modelo em rejeitar falsos positivos, uma característica vital para a confiabilidade de sistemas de diagnóstico automatizado.

A classe Intérfase destacou-se com o melhor desempenho global (F1-Score de 0,93). A alta Sensibilidade (91,7%) aliada a uma Especificidade de 95,5% mostram que, neste conjunto de validação, o modelo teve um bom desempenho na distinção entre células que não estão em divisão e células em divisão.

Este resultado é fundamental para a validade do índice mitótico, pois assegura que a contagem total de células não seja inflacionada por falsos positivos.

A Prófase, que historicamente apresenta desafios de identificação, obteve uma sensibilidade elevada de 89,4% e uma especificidade de 95,6%. Esse resultado supera limitações de treinamentos anteriores, sugerindo que a etapa de *fine-tuning* permitiu à rede neural distinguir sutilezas da condensação cromossômica inicial, mantendo um comportamento conservador que gera poucos alarmes falsos.

Além disso, a Telófase obteve a maior taxa de detecção do conjunto (Sensibilidade de 95%), capturando a quase totalidade das células reais nesta fase. As fases Metáfase e Anáfase apresentaram as maiores especificidades (98,9% e 98,4%, respectivamente), confirmando que,

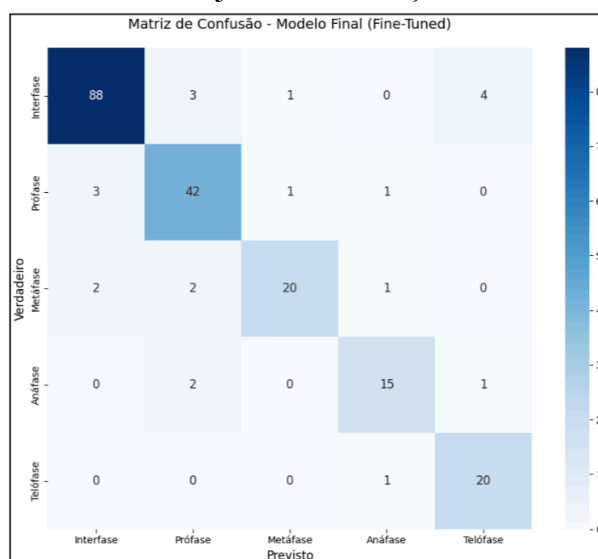
quando o modelo identifica estas fases críticas, a probabilidade de acerto é alta.

Embora erros residuais tenham sido observados entre fases mitóticas adjacentes, como Prófase e Metáfase, esses equívocos não comprometem o objetivo central da aplicação. O cálculo do índice mitótico depende fundamentalmente da correta distinção entre células em divisão e células em interfase. Nesse contexto, o elevado desempenho obtido para a classe Interfase (F1-Score de 0,93) assegura a confiabilidade da estimativa do índice mitótico, uma vez que minimiza a ocorrência de falsos positivos na contagem de células não divisórias. Assim, mesmo diante da complexidade inerente à segmentação fina das fases mitóticas, o sistema demonstra robustez para a finalidade principal da análise citogenética automatizada.

5.2 INTERPRETAÇÃO DA MATRIZ DE CONFUSÃO

Para investigar a natureza dos erros de classificação, analisou-se a Matriz de Confusão na Figura 16. A distribuição dos erros não é estocástica, mas sim biologicamente coerente.

Figura 16 - Matriz de Confusão do modelo no conjunto de validação.



Fonte: Autores (2025).

A matriz de confusão revela que a principal fonte de erro observada em treinamentos anteriores é a confusão entre Prófase e Interfase que nesta foi drasticamente reduzida. Observou-se que apenas 3 imagens de Prófase foram classificadas incorretamente como Interfase. Biologicamente, esse resultado indica que o modelo superou a ambiguidade visual entre a fase Interfase tardia e o início da condensação da cromatina, aprendendo a detectar padrões de textura nuclear sutis que definem o início da divisão.

As confusões entre as fases sequenciais finais também foram mitigadas. A dificuldade

em demarcar o limite entre Anáfase e Telófase, que anteriormente gerava trocas de rótulos frequentes, resultou em apenas um caso de classificação incorreta (uma Anáfase classificada como Telófase). Isso demonstra que a rede neural aprendeu a distinguir a separação dos polos e o início da descondensação.

Notavelmente, o modelo apresentou uma robustez significativa na distinção entre classes morfológicamente opostas. Na fase de Metáfase, por exemplo, os poucos erros existentes, 5 imagens, concentraram-se majoritariamente na confusão com Interfase ou Prófase, provavelmente devido a ângulos de captura onde a placa metafásica não estava perfeitamente visível, sem apresentar confusões graves com a Telófase.

5.3 ANÁLISE COMPARATIVA COM O ESTADO DA ARTE

Para contextualizar a relevância da ferramenta construída, os resultados foram confrontados com o estudo recente de Balagbis et al. (2024), que empregou a arquitetura YOLOv8 para a análise de células de *Allium cepa*.

Embora Balagbis et al. (2024) tenham reportado uma precisão média (mAP) de 88,3%, a abordagem restringiu-se a uma classificação binária: "células em divisão" ou "não divisão". O presente trabalho assumiu uma complexidade computacional superior ao implementar a classificação multiclasse, segregando as células em cinco fases distintas do ciclo celular.

Ao isolarmos a performance do modelo *EfficientNet-B0* na classe Interfase, que é análoga à classe "não divisão" do estado da arte, obteve-se um F1-Score de 0,93 (Tabela 1). Este dado corrobora que a ferramenta desenvolvida não apenas compete em performance com o estado da arte para a finalidade de cálculo do índice mitótico, mas oferece a vantagem de fornecer o diagnóstico detalhado das fases da mitose (interfase, prófase, metáfase, anáfase e telófase), enriquecendo substancialmente a análise citogenética.

A integração dos modelos na aplicação *web* permitiu operacionalizar o fluxo de análise. A interface foi projetada para funcionar como uma ferramenta de apoio à decisão, possibilitando ao pesquisador revisar as previsões automáticas. Desta forma, o sistema busca otimizar o tempo de análise quantitativa, mantendo o especialista no ciclo de validação para assegurar a precisão final do diagnóstico.

Considerando uma acurácia global de 89%, o erro residual do sistema situa-se em torno de 11%, valor que tende a ser reduzido após o ajuste fino e a validação assistida pelo especialista. Na prática laboratorial, a contagem manual do índice mitótico está sujeita a erros associados ao cansaço visual, à subjetividade na interpretação das fases e à variabilidade

intra-observador, especialmente em análises extensas. Dessa forma, o erro do sistema automatizado mostra-se comparável e potencialmente inferior ao erro humano acumulado ao longo de sessões prolongadas de análise, reforçando a viabilidade da ferramenta como apoio confiável à decisão, sem eliminar o papel do especialista.

6 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho atingiu seu objetivo ao desenvolver e validar um pipeline computacional híbrido para a automação da análise citogenética de células vegetais. A aplicação web desenvolvida confirmou a viabilidade prática da abordagem, oferecendo suporte à revisão das análises e reduzindo o tempo despendido em tarefas manuais de contagem, tanto em contextos de pesquisa quanto em atividades didáticas. As limitações observadas concentram-se em imagens com alta sobreposição celular ou artefatos de coloração, indicando a necessidade de maior padronização dos dados de entrada. Assim, este trabalho representa uma etapa inicial de um processo contínuo de aprimoramento.

Como trabalhos futuros, destaca-se a construção de um *dataset* autoral no Laboratório de Biotecnologia Ambiental e Genotoxicidade (BIOGEN/UNIFAL-MG), incluindo imagens padronizadas de *Allium cepa* e, posteriormente, de *Lactuca sativa*. A ampliação e qualificação desse acervo permitirão o refinamento dos modelos de segmentação e classificação, bem como a evolução da aplicação web para uso rotineiro por docentes e pesquisadores da instituição.

REFERÊNCIAS

- BALAGBIS, Rachel Anne B. et al. Classification of dividing and non-dividing cells in *Allium cepa* assay using YOLOv8. **Procedia Computer Science**, v. 251, p. 240–247, 2024.
- BOURN, Matthew D. et al. Evaluation of image analysis tools for the measurement of cellular morphology. **Frontiers in Cell and Developmental Biology**, v. 13, p. 1572212, 15 maio 2025.
- BRAMMER, Sandra Patussi; TONIAZZO, Claudia; POERSCH, Liane Balvedi. Corantes comumente empregados na citogenética vegetal. **Arquivos do Instituto Biológico**, v. 82, n. 0, 2015.
- DONOVAN, Taryn A. *et al.* Mitotic Figures—Normal, Atypical, and Imposters: A guide to identification. **Veterinary Pathology**, v. 58, n. 2, p. 243–257, mar. 2021.
- FROTA, Rafael Gonzalez *et al.* Citotoxicidade, genotoxicidade e mutagenicidade da infusão de *Plectranthus barbatus* – Lamiaceae (malva-santa) avaliada pelo sistema teste *Allium cepa*. **Revista de Ciências Médicas e Biológicas**, v. 18, n. 1, p. 67, 3 jul. 2019.
- JIANG, Hao *et al.* Deep learning for computational cytology: A survey. **Medical Image Analysis**, v. 84, p. 102691, fev. 2023.
- JIANG, Qingxin et al. HyFormer: a hybrid transformer-CNN architecture for retinal OCT image segmentation. **Biomedical Optics Express**, v. 15, n. 11, p. 6156, 1 nov. 2024.
- NICUȚĂ, Daniela *et al.* The *Allium cepa* Model: A review of its application as a cytogenetic tool for evaluating the biosafety potential of plant extracts. **Methods and Protocols**, v. 8, n. 4, p. 88, 2 ago. 2025.
- MEUTEN, D. J.; MOORE, F. M.; GEORGE, J. W. Mitotic count and the field of view area: time to standardize. **Veterinary Pathology**, v. 53, n. 1, p. 7–9, jan. 2016.
- ORCHID, Navya; PUTHANPURAYIL, Sathi. Factors affecting the assessment of mitotic count in histopathological sections of tumors: a study of interobserver and intraobserver variability. **International Journal of Research in Medical Sciences**, p. 762–765, 2016.
- STRINGER, Carsen et al. Cellpose: a generalist algorithm for cellular segmentation. **Nature Methods**, v. 18, n. 1, p. 100–106, jan. 2021.
- TAN, Mingxing; LE, Quoc V. EfficientNet: rethinking model scaling for convolutional neural networks. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 36., 2019, Long Beach. **Proceedings** [...]. Long Beach: PMLR, 2019. p. 6105-6114.