

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Igor Blanco Toneti

**Utilização da YOLO na Identificação de  
Deficiências de Macronutrientes em Folhas de  
Café**

**Uberlândia, Brasil**

**2025**

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Igor Blanco Toneti

**Utilização da YOLO na Identificação de Deficiências de  
Macronutrientes em Folhas de Café**

Trabalho de conclusão de curso apresentado  
à Faculdade de Computação da Universidade  
Federal de Uberlândia, como parte dos  
requisitos exigidos para a obtenção título de  
Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Thiago Pirola Ribeiro

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2025

*Este trabalho é dedicado a Jesus Cristo, aos meus pais, meu irmão, familiares e amigos, que sempre estiveram ao meu lado em todas as situações.*

# Agradecimentos

Agradeço ao Prof. Dr. Thiago Pirola Ribeiro pela oportunidade e orientação dedicada, à meus Pais e meu irmão Pedro pelo apoio incondicional e à meus amigos Dahlan, Felipe, Lucas, Mateus, Thiago e Vinicio pela parceria durante todo meu trajeto.

“*Nada estará perdido enquanto estivermos em busca.*”

*(S. Agostinho de Hipona)*

# Resumo

O café representa uma das culturas economicamente mais importantes do mundo, sendo a América do Sul o principal produtor, seguido da Ásia e Oceania. A produtividade e qualidade do café dependem significativamente de alguns aspectos como a qualidade do solo, influenciado pela disponibilidade de nutrientes, como também do clima. Deficiências de macronutrientes como cálcio (Ca), magnésio (Mg), nitrogênio (N), fósforo (P) e potássio (K) e enxofre (S) podem causar sintomas visíveis nas folhas de café, levando à redução do rendimento e da qualidade se não forem tratadas prontamente. Os métodos tradicionais para diagnosticar deficiências de nutrientes dependem de inspeção manual e análise laboratorial, o que pode ser demorado, caro e sujeito a erro humano.

Avanços recentes em aprendizado profundo e visão computacional oferecem uma alternativa promissora para automatizar diagnósticos de saúde de plantas. O modelo de detecção de objetos “You Only Look Once” (YOLO) demonstrou alta precisão e desempenho em tempo real em várias aplicações agrícolas, sendo elas detecção de culturas e pragas. Neste estudo, utilizou-se da oitava versão do framework YOLO para detectar deficiências de macronutrientes em folhas de café com base em sintomas visuais. O YOLOv8 melhora seus predecessores com recursos aprimorados de detecção de objetos, melhor generalização e maior eficiência computacional, tornando-o adequado para essa tarefa.

Nos testes realizados com o dataset BDICAFE, a proposta apresentada nesse trabalho obteve acurárias de até 95,8%, conforme os testes realizados. Isso demonstra que a proposta pode servir como uma ferramenta valiosa para agricultores e agrônomos, permitindo a detecção precoce e o manejo direcionado de nutrientes para otimizar a produção de café.

**Palavras-chave:** Visão Computacional, YOLO, BDICAFE, Macronutrientes, Café.

# Lista de ilustrações

Figura 1 – Imagens de folhas de café com suas respectivas deficiências de macro-nutrientes. . . . .	17
Figura 2 – Passos do processamento de imagens. . . . .	18
Figura 3 – Exemplos da aplicação de pré-processamento na icônica imagem de Albert Einstein. . . . .	20
Figura 4 – Imagem abstrata de uma RNC simples com camadas convolucionais, <i>pooling</i> e totalmente conectadas. . . . .	21
Figura 5 – Exemplo da operação de convolução, em que um filtro (kernel) realiza operações elemento a elemento em uma imagem de entrada, resultando assim em um mapa de características. . . . .	22
Figura 6 – Exemplo do uso da operação <i>padding</i> . . . . .	23
Figura 7 – Exemplo da operação de <i>max pooling</i> , onde em cada espaço é selecionado o pixel com maior valor e no final obtém-se uma imagem com dimensionalidade reduzida. . . . .	24
Figura 8 – Exemplo de 2 camadas totalmente conectadas, observa-se os neurônios da camada $n_1$ conectando com os neurônios da camada $n_2$ . . . . .	26
Figura 9 – Imagem abstrata de uma arquitetura U-net. . . . .	26
Figura 10 – Exemplo da técnica convolução transposta, onde o objetivo é aumentar o mapa de características. . . . .	27
Figura 11 – Imagem abstrata de uma arquitetura U <sup>2</sup> -net. . . . .	28
Figura 12 – Arquitetura do modelo YOLOv8 da ultralytics. . . . .	30
Figura 13 – Todas as classes definidas no BDICAFE. As classes completo, nitrogênio e potássio, em destaque, foram as retiradas conforme os experimentos. . . . .	34
Figura 14 – Exemplo de um resultado da aplicação da biblioteca <code>rembg</code> para uma imagem do Dataset BDICAFE. . . . .	35
Figura 15 – Exemplo do filtro de contraste aplicado sobre uma imagem do Dataset BDICAFE. . . . .	36
Figura 16 – Resultados do treinamento utilizando todas as imagens do Dataset BDICAFE. . . . .	37
Figura 17 – Matriz de confusão dos testes realizados com todas as imagens do Dataset BDICAFE. Demonstra a dificuldade do modelo de fazer previsões, sua acurácia de 55% seria apenas uma previsão aleatória. . . . .	38
Figura 18 – Resultados do treinamento utilizando imagens selecionadas do Dataset BDICAFE. . . . .	40
Figura 19 – Matriz de confusão dos testes realizados com as imagens selecionadas do Dataset BDICAFE. . . . .	41

Figura 20 – Resultados do treinamento utilizando imagens selecionadas do Dataset BDICAFE aplicadas sob um filtro de aumento de contraste. . . . .	41
Figura 21 – Matriz de confusão dos testes realizados com as imagens selecionadas do Dataset BDICAFE aplicadas sob um filtro de aumento de contraste. . . . .	42
Figura 22 – Resultados do treinamento utilizando imagens selecionadas do Dataset BDICAFE, aplicadas sob um filtro de remoção de fundo. . . . .	42
Figura 23 – Matriz de confusão dos testes realizados com as imagens selecionadas do Dataset BDICAFE, aplicadas sob um filtro de remoção de fundo. . . . .	43
Figura 24 – Resultados do treinamento utilizando imagens selecionadas do Dataset BDICAFE. . . . .	57
Figura 25 – Matriz de confusão dos testes realizados com as imagens selecionadas do Dataset BDICAFE. . . . .	58
Figura 26 – Resultados do treinamento utilizando imagens selecionadas do Dataset BDICAFE aplicadas sob um filtro de aumento de contraste. . . . .	58
Figura 27 – Matriz de confusão dos testes realizados com as imagens selecionadas do Dataset BDICAFE aplicadas sob um filtro de aumento de contraste. . . . .	59
Figura 28 – Resultados do treinamento utilizando imagens selecionadas do Dataset BDICAFE, aplicadas sob um filtro de remoção de fundo. . . . .	59
Figura 29 – Matriz de confusão dos testes realizados com as imagens selecionadas do Dataset BDICAFE, aplicadas sob um filtro de remoção de fundo. . . . .	60
Figura 30 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE com exclusão da classe completo. . . . .	61
Figura 31 – Matriz de confusão dos testes realizados com as imagens selecionadas do dataset BDICAFE com exclusão da classe completo. . . . .	62
Figura 32 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE, sem a classe saudável, aplicadas sob um filtro de aumento de contraste. . . . .	62
Figura 33 – Matriz de confusão dos testes realizados com as imagens selecionadas do dataset BDICAFE, sem as classe saudável, aplicadas sob um filtro de aumento de contraste. . . . .	63
Figura 34 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE, sem a classe saudável, aplicadas sob um filtro de remoção de fundo. . . . .	64
Figura 35 – Matriz de confusão dos testes realizados com as imagens selecionadas do dataset BDICAFE, sem as classe saudável, aplicadas sob um filtro de remoção de fundo. . . . .	65
Figura 36 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE com exclusão da classe completo e nitrogênio. . . . .	65

Figura 37 – Matriz de confusão dos testes realizados com as imagens selecionadas do dataset BDICAFE com exclusão da classe completo e nitrogênio. . . . .	66
Figura 38 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE, sem as classes saudável e nitrogênio, aplicadas sob um filtro de aumento de contraste. . . . .	66
Figura 39 – Matriz de confusão dos testes realizados com as imagens selecionadas do dataset BDICAFE sem as classe saudável e nitrogênio aplicadas sob um filtro de aumento de contraste. . . . .	67
Figura 40 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE, sem as classes saudável e nitrogênio, aplicadas sob um filtro de remoção de fundo. . . . .	67
Figura 41 – Matriz de confusão dos testes realizados com as imagens selecionadas do dataset BDICAFE, sem as classe saudável e nitrogênio, aplicadas sob um filtro de remoção de fundo. . . . .	68
Figura 42 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE com exclusão da classe completo, nitrogênio e potássio. . . . .	68
Figura 43 – Matriz de confusão dos testes realizados com as imagens selecionadas do dataset BDICAFE com exclusão da classe completo, nitrogênio e potássio. . . . .	69
Figura 44 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE, sem as classes saudável, nitrogênio e potássio, aplicadas sob um filtro de aumento de contraste. . . . .	69
Figura 45 – Matriz de confusão dos testes realizados com as imagens selecionadas do dataset BDICAFE sem as classe saudável, nitrogênio e potássio aplicadas sob um filtro de aumento de contraste. . . . .	70
Figura 46 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE, sem a classe saudável, nitrogênio e potássio, aplicadas sob um filtro de remoção de fundo. . . . .	71
Figura 47 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE, sem as classes saudável, nitrogênio e potássio, aplicadas sob um filtro de remoção de fundo. . . . .	72

# Lista de tabelas

Tabela 1 – Versões dos modelos YOLO e suas contribuições.	29
Tabela 2 – Representação da matriz de confusão para um problema de classe binária.	32
Tabela 3 – Tabela com a Acurácia para os testes realizados com o Dataset contendo todas as classes.	40
Tabela 4 – Tabela com a Acurácia para os testes realizados com o Dataset sem a classe completa.	44
Tabela 5 – Tabela com a acurácia dos testes realizados com o Dataset sem a classe completa e nitrogênio	46
Tabela 6 – Tabela com a acurácia dos testes realizados com o Dataset sem a classe completa, nitrogênio e potássio	47
Tabela 7 – Tabela com a Acurácia dos testes realizados com o Dataset sem filtros adicionados	48
Tabela 8 – Tabela com a Acurácia dos testes realizados com o Dataset com aumento de contraste	48
Tabela 9 – Tabela com a Acurácia dos testes realizados com o Dataset com remoção de fundo	49
Tabela 10 – Tabela com as melhores Acurárias dos testes realizados.	50

# Lista de abreviaturas e siglas

**FN** Falso Negativo

**FP** Falso Positivo

**HSV** Matiz, Saturação e Valor - *Hue, Saturation and Value*

**LAB** Luminosidade, Vermelho-Verde e Azul-Amarelo - *Lightness, Red-Green, and Blue-Yellow*

**R-CNN** Redes Neurais Convolucionais baseadas em Região - *Region-based Convolutional Neural Networks*

**RGB** Vermelho, Verde e Azul - *Red, Green and Blue*

**RNA** Redes Neurais Artificiais - *Artificial Neural Network*

**RNC** Redes Neurais Convolucionais - *Convolutional Neural Network*

**VN** Verdadeiro Negativo

**VP** Verdadeiro Positivo

**YOLO** Você só olha uma vez - *You Only Look Once*

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
<b>1.1</b>	<b>Objetivos</b>	<b>14</b>
<b>1.2</b>	<b>Organização da Monografia</b>	<b>14</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>16</b>
<b>2.1</b>	<b>Deficiências de minerais no café</b>	<b>16</b>
<b>2.2</b>	<b>Processamento de Imagens</b>	<b>17</b>
2.2.1	Brilho	19
2.2.2	Contraste	19
2.2.3	Remoção de fundo	20
<b>2.3</b>	<b>Redes Neurais Convolucionais</b>	<b>20</b>
2.3.1	Convolução	21
2.3.1.1	<i>Kernel</i>	22
2.3.1.2	<i>Stride</i>	22
2.3.1.3	<i>Padding</i>	23
2.3.2	<i>Pooling</i>	24
2.3.3	Normalização em lote	24
2.3.4	Camadas totalmente conectadas	25
2.3.5	U-net e U <sup>2</sup> -Net	25
2.3.6	YOLO	27
2.3.6.1	Arquitetura YOLOv8	29
2.3.6.1.1	<i>Backbone</i>	29
2.3.6.1.2	<i>Neck</i>	30
2.3.6.1.3	<i>Head</i>	31
2.3.7	Métodos de amostragem	31
2.3.8	Métricas de avaliação	31
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>33</b>
<b>3.1</b>	<b>Materiais e Métodos</b>	<b>33</b>
3.1.1	Banco de imagens	33
3.1.2	YOLOv8	34
3.1.3	Remoção de Fundo	34
3.1.4	Contraste	35
<b>3.2</b>	<b>Experimentos</b>	<b>36</b>
3.2.1	1º Experimento	39
3.2.2	2º Experimento	43

3.2.3	3º Experimento . . . . .	44
3.2.4	4º Experimento . . . . .	46
3.2.5	Análise das Acurárias dos Experimentos . . . . .	47
3.2.5.1	Acurácia do Dataset sem filtro . . . . .	47
3.2.5.2	Acurácia do Dataset com aumento de contraste . . . . .	48
3.2.5.3	Acurácia do Dataset com remoção de fundo . . . . .	49
3.2.5.4	Melhores Acurárias . . . . .	49
<b>4</b>	<b>CONCLUSÃO . . . . .</b>	<b>51</b>
<b>4.1</b>	<b>Trabalhos Futuros . . . . .</b>	<b>52</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>53</b>
	<b>APÊNDICES</b>	<b>55</b>
	<b>APÊNDICE A – PARÂMETROS DA YOLOV8 . . . . .</b>	<b>56</b>
	<b>APÊNDICE B – TABELAS E GRÁFICOS DOS EXPERIMENTOS</b>	<b>57</b>
<b>B.1</b>	<b>1º Experimento . . . . .</b>	<b>57</b>
<b>B.2</b>	<b>2º Experimento . . . . .</b>	<b>60</b>
<b>B.3</b>	<b>3º Experimento . . . . .</b>	<b>60</b>
<b>B.4</b>	<b>4º Experimento . . . . .</b>	<b>61</b>

# 1 Introdução

O Brasil é o maior produtor de café, abastecendo 30% do mercado global e em relação ao consumo está atrás apenas dos Estados Unidos. Anualmente o consumo de café vem aumentando, como também compradores nacionais e internacionais elevaram seus padrões, exigindo grãos de café de maior qualidade e com características específicas, além de fiscalizações mais rigorosas para produção. Como resultado, os cafeicultores brasileiros enfrentam o desafio de manter altos níveis de produção e, ao mesmo tempo, atender a essas expectativas de maior qualidade do mercado (CHALFOUN; REIS, 2010).

No ambiente de cultivo, o agricultor deve estar preparado para enfrentar diversos obstáculos que podem colocar sua cultura em risco, sendo os mais comuns o clima, pragas e deficiência de nutrientes no solo. As variações de abordagem para combater cada problema são diferentes e dependem muito do momento em que são usadas, levando o agricultor a ter gastos maiores se não agir de maneira antecipada.

Uma importante abordagem é a detecção precoce de deficiências de nutrientes, pois é crucial para manter as plantações saudáveis e garantir o rendimento ideal. Quando as plantas carecem de nutrientes essenciais como cálcio, magnésio, nitrogênio, fósforo, potássio e enxofre, elas apresentam sintomas como clorose, crescimento atrofiado e deformação das folhas. Se não forem tratadas, essas deficiências podem levar a estresse severo nas plantas, produtividade reduzida e, em casos extremos, morte da planta e quebra de safra. Por estar diretamente associado ao crescimento da planta e desenvolvimento de frutos, a detecção precoce de deficiência de macronutrientes evita danos irreversíveis nas plantas como na economia. A intervenção no tempo correto permite que as plantas recebam cuidados específicos evitando medidas corretivas de emergência, que geralmente são mais caras e menos eficazes. Outro ponto importante é a suscetibilidade a doenças, plantas fracas são mais suscetíveis à doenças o que leva a maiores cuidados e mais gastos para o produtor. Portanto, ao alavancar técnicas de detecção precoce, os agricultores podem tomar medidas proativas para corrigir deficiências antes que se tornem graves, melhorando a produtividade, reduzindo o desperdício e garantindo a segurança alimentar.

Nos últimos anos, a sinergia entre visão computacional e agricultura tem impulsionado avanços substanciais, especialmente no contexto da agricultura de precisão e na automação de processos agrícolas, afetando tanto o agricultor como indústrias. Um dos principais catalisadores dessa transformação é a evolução contínua da arquitetura You Only Look Once (YOLO), uma família de detectores de objetos que se destaca por sua elevada eficiência computacional e precisão em tempo real.

A primeira versão do YOLO, proposta por Joseph Redmon em 2016 [Redmon](#)

(2016), representou um marco disruptivo ao consolidar o processo de detecção em uma única etapa, ou seja com apenas uma passagem pela rede, contrastando com os paradigmas tradicionais de dois estágios, como os utilizados por Redes Neurais Convolucionais baseadas em Região (R-CNN) e suas variantes. Essa abordagem permite à rede dividir a imagem de entrada em uma malha de células e, de forma paralela, inferir as caixas delimitadoras, também conhecidas como *bouding boxes*, e as probabilidades de classe, o que resulta em um desempenho de inferência significativamente mais rápido sem perdas expressivas em acurácia.

Com a evolução das versões — YOLOv1 ao YOLOv8 — houve uma progressiva incorporação de aprimoramentos arquiteturais, estratégias avançadas de treinamento e técnicas de otimização específicas, como detecção de alvos de pequena escala, oclusões parciais e variações nos domínios dos dados. Cada nova iteração da família YOLO tem buscado mitigar limitações anteriores (ALIF; HUSSAIN, 2024).

Por se tratar de um problema do campo da visão computacional, pois deficiências nesses macronutrientes se manifestam visivelmente nas folhas e afetam negativamente a saúde geral, o uso da rede YOLOv8 para detectar deficiências em folhas de café poderá alcançar alta precisão, velocidade, eficiência e generalização na identificação e classificação de sintomas de deficiências devido aos seus recursos avançados de detecção de objetos, extração robusta de características e processamento em tempo real.

## 1.1 Objetivos

O estudo tem como objetivo otimizar técnicas de detecção de deficiências em folhas de plantas de café, contribuindo assim para uma classificação eficiente que ajude profissionais agrônomos em suas tarefas cotidianas.

Portanto, pretende-se:

- Avaliar o framework YOLOv8 para classificação de deficiência de macronutrientes em folhas de planta de café;
- Analisar e interpretar filtros na classificação de plantas de folha de café;

## 1.2 Organização da Monografia

Neste capítulo, foram apresentadas as considerações iniciais, motivação hipótese, problema e objetivos para o desenvolvimento do método proposto. O restante deste trabalho está divididos em revisão bibliográfica, desenvolvimento, resultados e conclusão. No Capítulo 2 da revisão bibliográfica é apresentado a teoria e o estado da arte de tópicos relacionados problemas de deficiência de macronutrientes na folha de café,

processamento digital de imagens e métodos de extração de características e classificação. No Capítulo 3 de desenvolvimento são apresentados os materiais e métodos utilizados para o desenvolvimento deste projeto, sendo eles: descrição do banco de imagens utilizado, descrição de técnicas utilizadas para processar imagens, descrição das técnicas utilizadas nas etapas de extração de características e classificação assim como as métricas empregadas para avaliação do método proposto. Além disso, apresenta os resultados obtidos em cada etapa empregada, bem como os resultados quantitativos dos experimentos e uma breve discussão e análise e por último o Capítulo 4 em que apresenta as considerações e pesquisas futuras do projeto.

## 2 Revisão Bibliográfica

Este capítulo explora o impacto das deficiências de macronutrientes nas plantas de café e como a visão computacional pode auxiliar na identificação dessas deficiências. Macronutrientes são essenciais para o crescimento saudável das plantas, e suas deficiências podem levar a sintomas visíveis, como clorose, murchamento, mudanças de cor e necrose. Os métodos tradicionais de diagnóstico desses problemas podem ser demorados e exigem conhecimento especializado, além do envio de amostras para análises laboratoriais. No entanto, os avanços em visão computacional permitem a detecção automatizada, precisa e eficiente de deficiências de nutrientes por meio de análise de imagem e aprendizado de máquina. Este capítulo se aprofunda nos sintomas de deficiências comuns de macronutrientes e como as tecnologias de visão computacional podem revolucionar a agricultura de precisão na cafeicultura.

### 2.1 Deficiências de minerais no café

O solo serve como meio de onde as plantas obtêm os minerais essenciais para seu desenvolvimento e quando esse meio não contém ou não disponibiliza quantidades suficientes de nutrientes, algo que pode ser constatado pela análise química do solo, as plantas não terão suas necessidades nutricionais plenamente atendidas, resultando em problemas característicos definidos visivelmente devido ao estresse da planta (FAQUIN, 2002).

Em relação às plantas de café os níveis de nutrientes dependem de vários fatores, como o local em que a planta é cultivada, a época do ano, a idade da planta e as características genéticas, isso afetará o crescimento da planta, podendo resultar em crescimento mais lento e menor produtividade ou um crescimento rápido e maior produtividade. Portanto, é importante garantir que a planta receba nutrientes suficientes, pois a nutrição mineral adequada é crucial para a produtividade da planta de café, aumentando a eficiência metabólica da planta (FAQUIN, 2002).

Deficiências de minerais no café se manifestam por meio de vários sintomas nas folhas, impactando a saúde e a produtividade da planta. Uma visão geral das deficiências de macronutrientes comuns, seus sintomas e potenciais consequências (Figura 1):

- **Cálcio (CA):** Murchamento das folhas, colapso do pecíolo; clorose marginal; manchas nos frutos, morte das gemas;
- **Potássio (K):** Clorose e depois necrose das pontas e margens;

- **Magnésio (Mg):** Clorose internerval seguida ou não da cor vermelho-roxa;
- **Nitrogênio (N):** Clorose em geral uniforme;
- **Fósforo (P):** Cor verde azulada com ou sem amarelecimento das margens;
- **Enxofre (S):** Clorose geralmente uniforme;

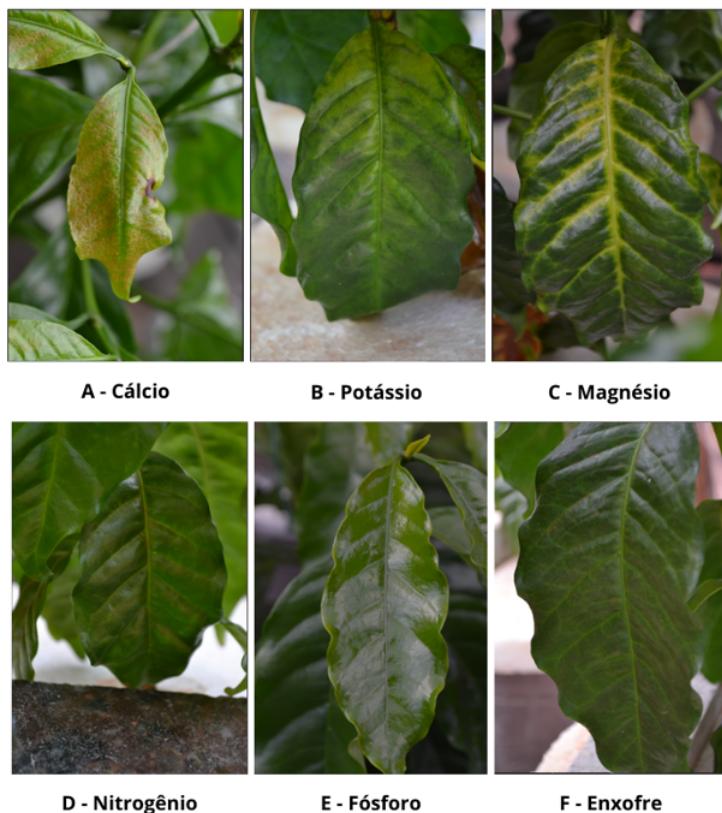


Figura 1 – Imagens de folhas de café com suas respectivas deficiências de macronutrientes.

Fonte: Dataset BDICAFE ([subseção 3.1.1](#)).

## 2.2 Processamento de Imagens

O processamento de imagens é um conjunto de passos em que realiza-se operações em imagens para melhorá-las, extrair informações importantes ou transformá-las em um formato para o uso final, humanamente interpretável. O processamento de imagens envolve analisar, manipular e interpretar imagens usando técnicas matemáticas e computacionais sendo muito utilizado em campos como imagens médicas, robótica, sensoriamento remoto e entretenimento.

Os passos do processamento de imagens (Figura 2) podem ser divididos em:

- **Aquisição da imagem** : Adquirir as imagens usando câmeras, scanners, sensores, raio X, ultrassom, etc;
- **Pré-processamento** : Melhorar a qualidade da imagem reduzindo ruído, ajustando brilho, contraste, removendo fundo, etc;
- **Segmentação** : Divisão da imagem em regiões importantes;
- **Extração de características** : Identificar partes importantes das imagens, como cantos, formas e texturas;
- **Classificação** : Análise e interpretação das características extraídas;

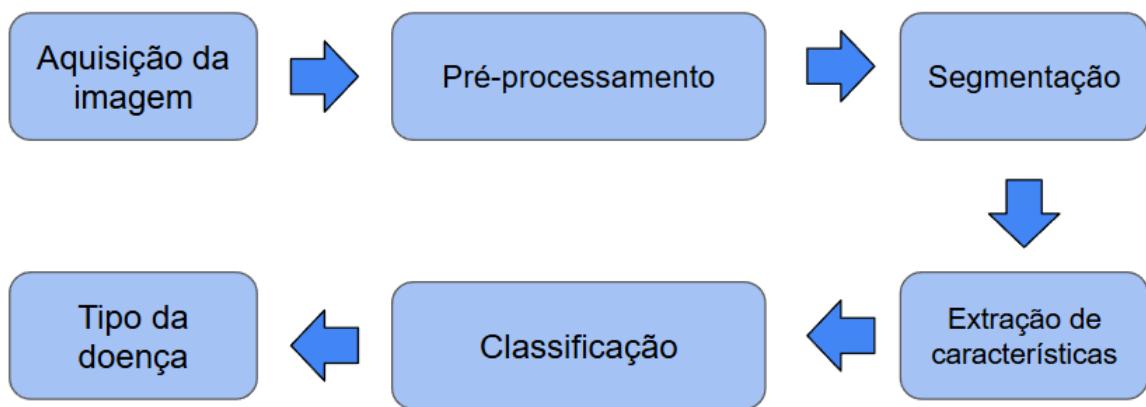


Figura 2 – Passos do processamento de imagens.

Fonte: Modificado de [Hungilo, Emmanuel e Emanuel \(2019\)](#).

Ocasionalmente, técnicas de pré-processamento podem ser implementadas para localizar, gerenciar e reduzir artefatos típicos em imagens. Técnicas básicas de processamento de imagens podem ser usadas junto com métodos mais avançados para excluir falsos positivos e reduzir erros de previsão em nível de pixel.

Para realização de algumas técnicas de pré-processamento faz-se necessária a modificação no espaço de cores das imagens. O espaço de cores em computação é um modelo matemático que define a forma como as cores são representadas em um sistema digital. Ele determina como as cores são combinadas e interpretadas para garantir consistência entre diferentes dispositivos. Alguns exemplos de espaços de cores são **RGB**, **HSV** e **LAB**. Segundo [Loesdau, Chabrier e Gabillon \(2014\)](#), o espaço de cor RGB consiste nas três cores primárias, vermelho, azul e verde, sendo baseado na teoria das cores. Atualmente a maioria dos aparelhos utiliza esse padrão para suas aplicações. De certo modo o modelo RGB possui pontos negativos em relação a segmentação de imagens, isso se deve porque mesmo embora o espaço RGB corresponda ao processamento biológico

de cores no sistema visual humano, ele não parece se correlacionar com a diferenciação perceptual humana entre cores. Outro espaço de cores é o HSV, introduzido por [Smith \(1978\)](#), que busca a percepção de matiz, saturação e valor. A matiz é uma posição no espectro de cores, saturação pode-se dizer que é a intensidade ou pureza da cor, quanto mais perto do cinza mais dessaturada é a cor e por final o valor é a luminosidade. O espaço de cor CIELAB<sup>1</sup>, normalmente denominado L\*a\*b ou [LAB](#), é um padrão internacional projetado para uniformidade perceptual. A intenção é que as diferenças de cores que um humano percebe como iguais correspondam a distâncias euclidianas iguais no espaço [LAB](#), sendo assim um dos dois sistemas de coordenadas mais recomendados para comparar diferenças de cores ([SCHWARZ; COWAN; BEATTY, 1987](#)).

Segundo [Gonzalez \(2009\)](#), dentre as diversas técnicas de pré-processamento, algumas mais comuns são: brilho, contraste e remoção de fundo.

### 2.2.1 Brilho

No processamento de imagens, brilho se refere à intensidade percebida da luz em uma imagem. É um atributo escalar que descreve o quão brilhante ou escura é uma imagem. O ajuste do brilho normalmente parte do princípio de aumentar ou diminuir de maneira uniforme os valores de intensidade de todos os pixels da imagem.

O principal impacto da mudança do brilho na imagem ([Figura 3 - B](#)) é quanto maior o brilho mais clara é a imagem, do mesmo modo, quanto menor o brilho mais escura. Deve-se pontuar que, diferentemente do contraste que realça a diferença das partes mais claras e escuras da imagem o brilho por sua vez altera o nível geral de luz.

### 2.2.2 Contraste

O contraste se refere à diferença de intensidade ou luminância entre as partes mais claras e mais escuras de uma imagem. Ele determina o quão bem os detalhes na imagem se destacam contra o fundo, o com maior contraste tornando os detalhes mais distintos e menor contraste tornando-os menos perceptíveis ([Figura 3 - C](#)).

Contraste é uma medida do intervalo de valores de intensidade de pixel em uma imagem. Alto contraste melhora a nitidez e os detalhes, tornando as bordas e transições mais pronunciadas, enquanto baixo contraste resulta em uma imagem com características menos distinguíveis.

<sup>1</sup> Definido pela *International Commission on Illumination* <<https://www.cie.co.at>>

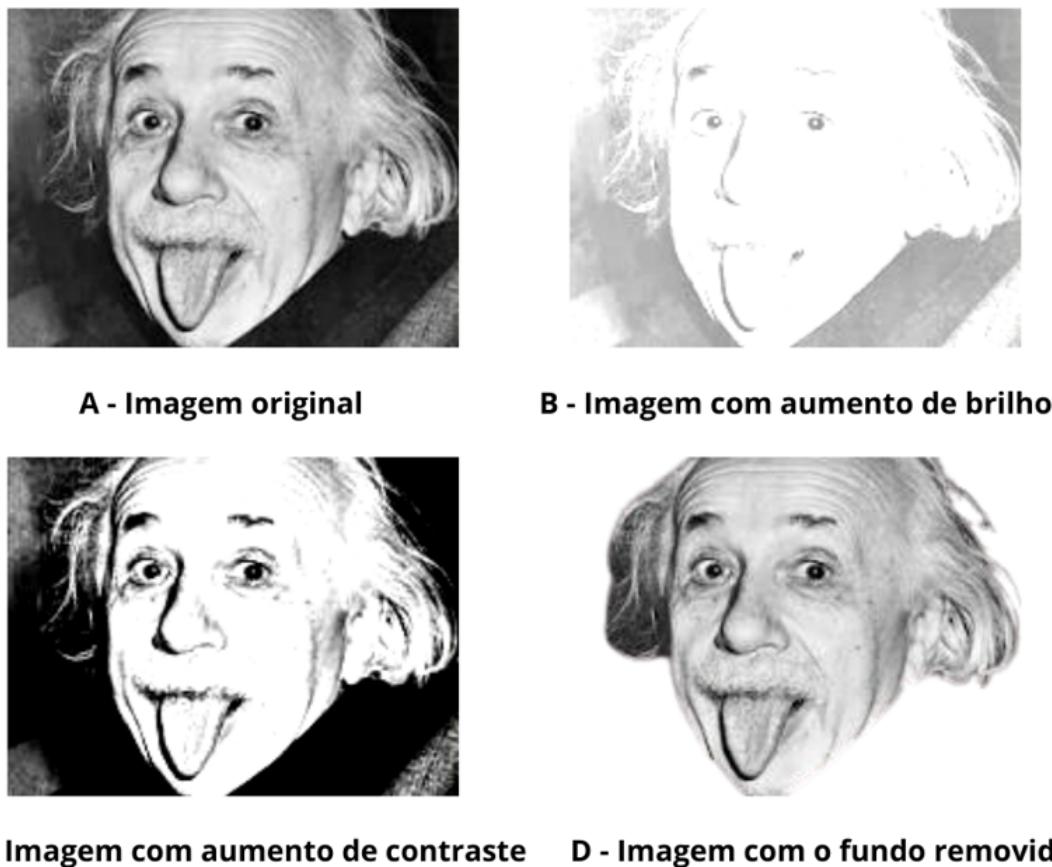


Figura 3 – Exemplos da aplicação de pré-processamento na icônica imagem de Albert Einstein.

Fonte: Figura modificada do [UOL](#).

### 2.2.3 Remoção de fundo

A técnica de remoção de fundo se refere ao isolamento do objeto do primeiro plano em relação ao fundo da imagem. Isso é comumente usado em detecção de objetos e envolve identificar e segmentar o assunto principal do resto da imagem. Exemplo de remoção de fundo pode ser encontrado na [Figura 3 - D](#).

Abordagens avançadas como segmentação semântica, por exemplo, utilizando os modelos U-net, podem ser exploradas para identificar objetos com precisão em imagens complexas.

## 2.3 Redes Neurais Convolucionais

O aprendizado profundo tem ganhado destaque nos últimos anos, sendo amplamente utilizado em pesquisas devido à sua capacidade de lidar com grandes volumes de dados de forma rápida e eficaz. Um dos principais usos dessas técnicas está no campo

do robótica e segurança, onde as Redes Neurais Artificiais (RNA) são aplicadas em tarefas como segmentação (identificação de objetos e regiões), extração de características e classificação de imagens, graças à sua versatilidade (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

As Redes Neurais Convolucionais (RNC) estão entre as abordagens mais utilizadas para a área de visão computacional devido à arquitetura dessas redes ser composta por camadas que combinadas de forma adequada produzem resultados específicos. Segundo Yamashita et al. (2018) as principais diferenças das RNCs e algoritmos de engenharia de características está principalmente na capacidade das RNCs aprenderem automaticamente, dispensando descritores parametrizados, para realizar tarefas como extração de características e classificação.

Um exemplo de arquitetura simplificada das RNCs pode ser observado na Figura 4. Os blocos mais comuns que constituem essas camadas são chamados de convolução, totalmente conectadas e *pooling*, sendo que este último não possui parâmetros ajustáveis (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

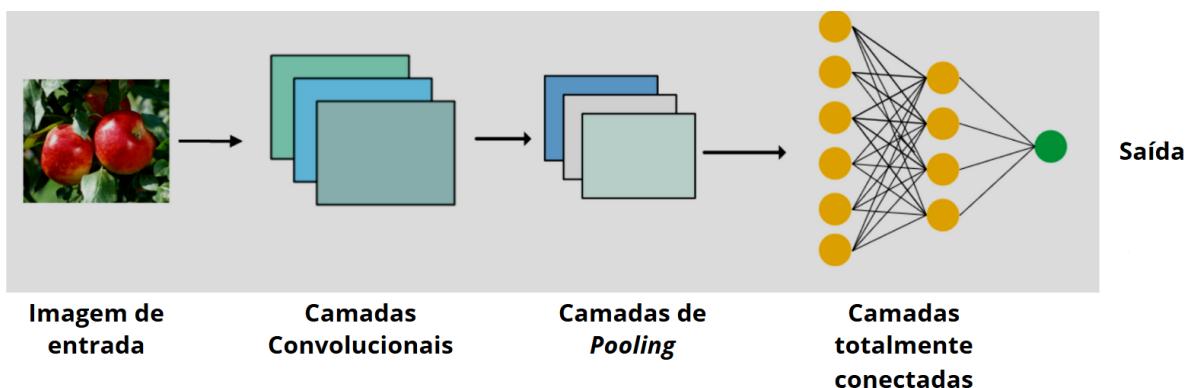


Figura 4 – Imagem abstrata de uma RNC simples com camadas convolucionais, *pooling* e totalmente conectadas.

Fonte: Traduzido de Alif e Hussain (2024).

### 2.3.1 Convolução

A convolução é uma operação matemática linear amplamente empregada na área de visão computacional, a convolução gera uma imagem ou um mapa de características a partir de uma imagem de entrada. Esse mapa não apenas indica a localização de objetos na imagem, mas também descreve suas propriedades.

A operação de convolução (Figura 5) é realizada com dois elementos, uma imagem ou mapa de entrada e um filtro, comumente referido como kernel, ambos representados por matrizes de valores discretos. O processo começa com a aplicação de uma operação linear,

elemento por elemento, entre a matriz da imagem e a do filtro, tal processo faz o filtro agir como um extrator de características que aprende com o tempo a extrair características importantes da imagem de entrada. O resultado da operação é outra matriz, conhecida como mapa de características ([YAMASHITA et al., 2018](#)).

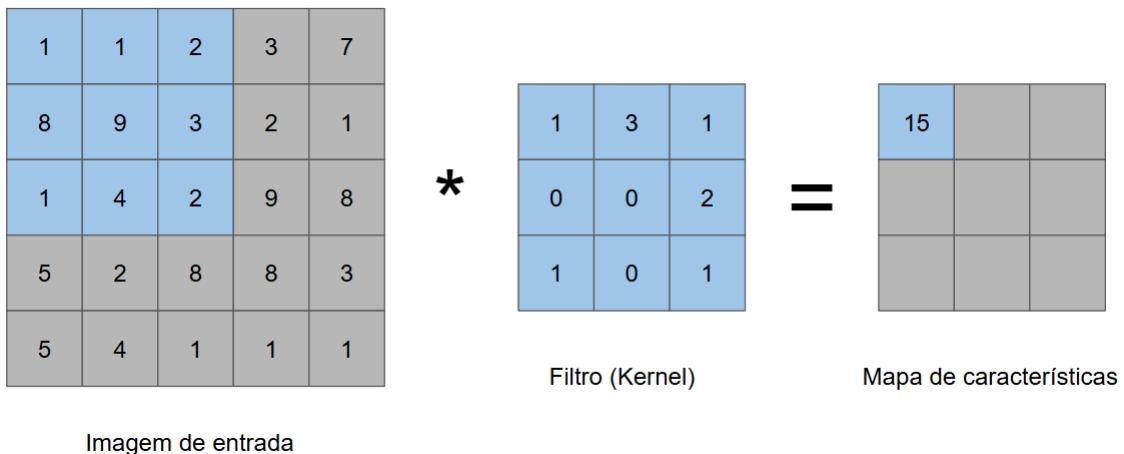


Figura 5 – Exemplo da operação de convolução, em que um filtro (kernel) realiza operações elemento a elemento em uma imagem de entrada, resultando assim em um mapa de características.

Fonte: Autoria própria

Nos modelos de visão computacional, as camadas iniciais são responsáveis por identificar características básicas, como cores, sombras e ângulos. Já as camadas mais profundas extraem padrões mais complexos, permitindo que o modelo forme uma compreensão abrangente da imagem analisada ([WU, 2017](#)).

Algumas operações de convolução e *pooling* são constituídas por *Kernel*, *Stride* e *Padding*.

### 2.3.1.1 Kernel

*Kernel* são matrizes, de tamanho igual ou menor a imagem de entrada, utilizada para operações de convolução. Essas matrizes “deslizam” pela imagem de entrada realizando operações de multiplicação elemento a elemento, a saída de cada operação realizada é chamada de mapa de características, onde cada mapa destaca características importantes da imagem.

### 2.3.1.2 Stride

A operação de *stride*, ou passo em português, se caracteriza pelo espaço, ou passo, que o kernel anda a cada operação realizada na imagem de entrada.

Aumentar o tamanho do *stride* tem como consequências:

- Reduzir custos computacionais.
- Capturar características de alto nível enquanto ignora características menores.

### 2.3.1.3 Padding

A operação *Padding*, ou preenchimento em português, constitui na adição de uma ou mais linhas e colunas de pixels na borda da imagem de entrada, dependendo do tamanho do kernel. Na [Figura 6](#) essa linha e coluna estão representados na cor laranja, exemplificando um *padding* para um kernel 3x3.

0	0	0	0	0	0	0
0	1	1	2	3	7	0
0	8	9	3	2	1	0
0	1	4	2	9	8	0
0	5	2	8	8	3	0
0	5	4	1	1	1	0
0	0	0	0	0	0	0

Imagen de entrada

Figura 6 – Exemplo do uso da operação *padding*.

Fonte: Autoria própria

O *padding* busca resolver dois problemas dentro das camadas de uma rede neural convolucional:

- Quanto maior a rede neural convolucional a dimensão espacial do imagem continua diminuindo levando a perda de informação;
- Os pixels das bordas da imagem de entrada sofrem menos operações convolucionais ou de *pooling* em relação aos pixels do centro da imagem, causando perda de informação das bordas caso alguma característica importante esteja localizada mais nos cantos da imagem.

### 2.3.2 Pooling

O *pooling* é uma etapa essencial em RNCs, pois é responsável por reduzir a dimensionalidade dos mapas de características. Esse processo faz um mapeamento dos valores da entrada em um tamanho menor, condensando o mapa de características de forma a preservar informações úteis e descartar as menos relevantes. O tipo de mapeamento depende da operação realizada, os tipos mais comuns são o *pooling* máximo onde é escolhido o pixel de maior valor pelo filtro (Figura 7), o *pooling* mínimo onde o menor é selecionado e *pooling* médio que é feita uma média dos valores dos pixels (YAMASHITA et al., 2018).

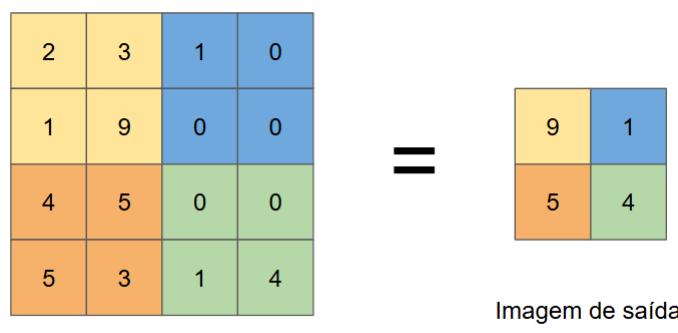


Figura 7 – Exemplo da operação de *max pooling*, onde em cada espaço é selecionado o pixel com maior valor e no final obtém-se uma imagem com dimensionalidade reduzida.

Fonte: Autoria própria

Segundo Yamashita et al. (2018), essa operação transforma a representação conjunta de características em dados mais significativos, ao mesmo tempo que introduz invariância a transformações espaciais e reduz a complexidade computacional para as camadas subsequentes. Isso ocorre ao eliminar parte das conexões entre as camadas convolucionais.

Os principais objetivos do *pooling* são:

- Diminuir o número de parâmetros ou pesos, reduzindo o custo computacional;
- Mitigar o *overfitting*, pois um método de redução de dimensionalidade eficiente deve extrair exclusivamente informações relevantes, descartando detalhes desnecessários;

### 2.3.3 Normalização em lote

A operação de normalização em lote é uma das principais técnicas utilizada no campo do aprendizado profundo, sendo uma técnica feita para estabilizar a distribuição

das entradas para uma camada específica durante o treinamento da RNC, sendo realizada antes da função de ativação da camada anterior. Isso é alcançado por meio da introdução de camadas adicionais que ajustam por meio da média e variância. Em seguida, essas entradas normalizadas são geralmente escaladas e deslocadas com base em parâmetros treináveis, permitindo preservar a capacidade expressiva do modelo.

A normalização de lotes busca:

- Estabilizar o processo de treinamento;
- Melhorar a generalização do modelo;
- Reduzir a sensibilidade dos pesos iniciais;
- Permitir maiores taxas de aprendizado;

### 2.3.4 Camadas totalmente conectadas

As camadas totalmente conectadas são um tipo de rede neural artificial que busca simular as estruturas e funções dos neurônios e suas conexões, elas processam e aprendem informações a partir dos dados. Normalmente nas arquiteturas das RNCs as camadas totalmente conectadas são implementadas após as operações de convolução e *pooling* no bloco final da rede. A partir do momento que todas as características são extraídas, isso depois de passar pelos blocos de convolução e *pooling*, ocorre um agrupamento delas e assim são usadas como entrada para as camadas totalmente conectadas. Dentro dessa parte, cada saída é conectada a entrada de outra camada com vários neurônios (Figura 8), e para cada ligação entre camadas ocorrem inúmeras operações onde são ajustadas as probabilidades para a classificação final (YAMASHITA et al., 2018).

Pode-se dizer que as camadas totalmente conectadas são representações de neurônios e que, no final do processamento, entregam um resultado. As camadas totalmente conectadas são responsáveis pela definição das probabilidades de cada amostra pertencer a uma determinada classe.

### 2.3.5 U-net e U<sup>2</sup>-Net

Segundo Siddique et al. (2021), o modelo U-net padrão (exemplificado na Figura 9), foi inicialmente desenvolvido para realizar segmentação, ou seja, detectar objetos em imagens e depois separá-los do plano de fundo. A U-net pode ser dividido em 2 partes principais: a contração e a expansão.

O primeiro caminho, referente a contração, é constituída de uma RNC básica, os blocos são divididos em 2 camadas de convoluções 3x3 seguida de uma função de ativação e um *polling* máximo, tal bloco é repetido inúmeras vezes. O segundo caminho,

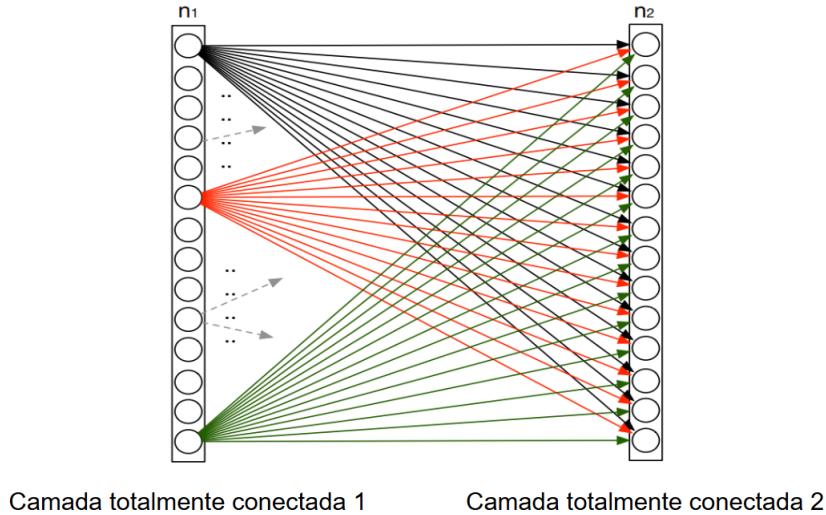


Figura 8 – Exemplo de 2 camadas totalmente conectadas, observa-se os neurônios da camada  $n_1$  conectando com os neurônios da camada  $n_2$ .

Fonte: Modificado de [Ma e Lu \(2017\)](#).

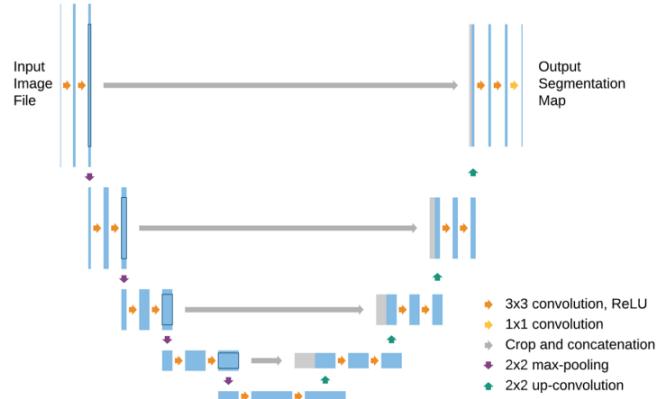


Figura 9 – Imagem abstrata de uma arquitetura U-net.

Fonte: [Siddique et al. \(2021\)](#)

referente a expansão, é o que torna as arquiteturas U-net destacado das demais RNCs, pois a cada estágio de forma gradual aumenta-se a imagem, ou mapa de características, usando uma técnica chamada convolução transposta ou deconvolução (Figura 10). Para realização da técnica é necessário uma imagem de entrada e um filtro, a técnica consiste na multiplicação elemento a elemento do filtro pela imagem aumentando assim o tamanho do mapa de características. Isso ajuda o modelo a reconstruir um mapa de segmentação detalhado a partir da representação compactada.

Dentro da estrutura da U-net existem concatenações que combinam informações dos resultados do caminho de expansão com o estágio do caminho de contração correspondente, formando-se de maneira imaginária uma arquitetura em formato da letra U dando assim origem ao nome U-net. Deve-se atentar que antes da concatenação o mapa de características é recortado devido as diferenças de tamanho dos mapas. A operação de concatenação permite que a rede retenha detalhes finos de camadas anteriores enquanto refina a segmentação.

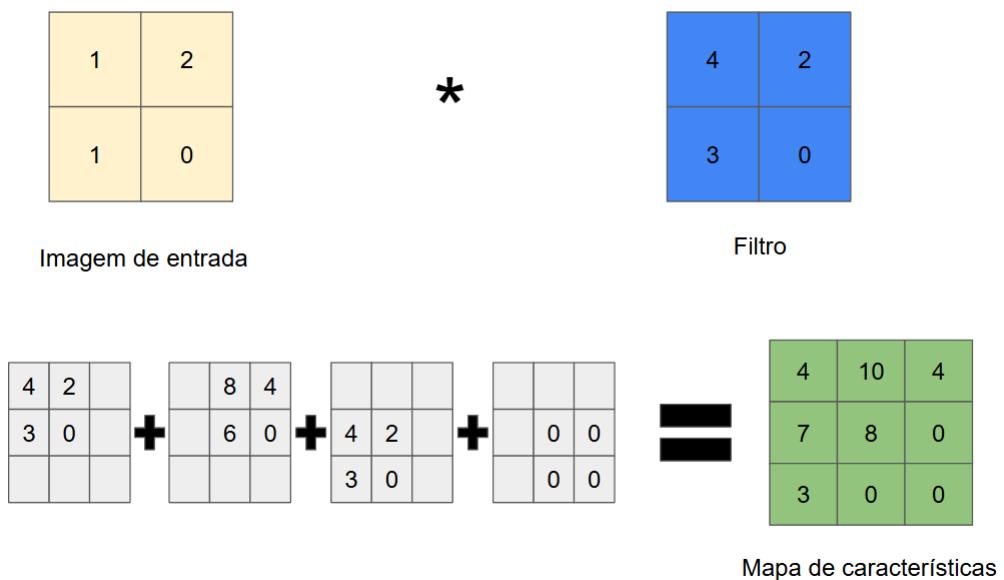


Figura 10 – Exemplo da técnica convolução transposta, onde o objetivo é aumentar o mapa de características.

Fonte: Autoria própria

Segundo [Jiang e Li \(2024\)](#) e [Qin et al. \(2020\)](#), a arquitetura tradicional das U<sup>2</sup>-nets são formadas principalmente por blocos U residuais, que substituem os blocos convolucionais comuns da U-net, imaginando-se uma estrutura recursiva em que cada bloco da U-net é substituído por uma estrutura similar a ela como demonstrado na [Figura 11](#), incorporando conexões residuais para mesclar localmente ou multi-escala por concatenação. Uma das características das U-nets recursivas é que quanto mais profunda for seus blocos mais poder computacional é necessário, afetando assim o tempo, elemento principal em tarefas em tempo real.

### 2.3.6 YOLO

Como afirmado por [Terven, Córdova-Esparza e Romero-González \(2023\)](#), dentro dos inúmeros algoritmos e modelos de classificação e detecção de objetos, os modelos da

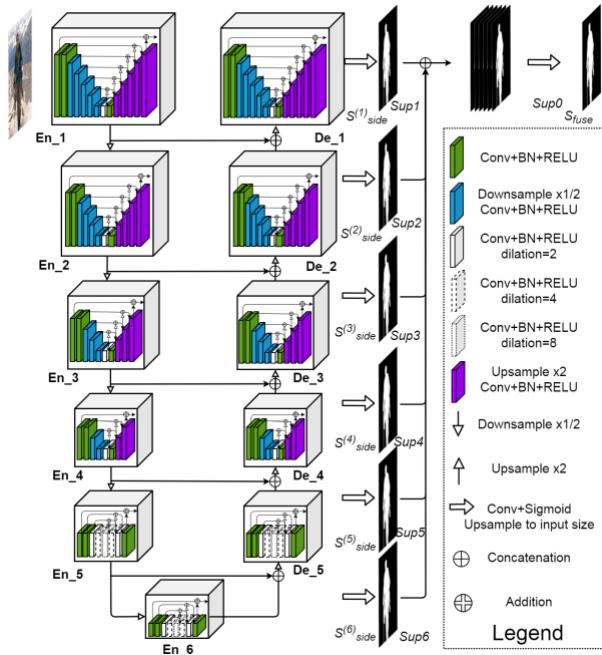


Figura 11 – Imagem abstrata de uma arquitetura U<sup>2</sup>-net.

Fonte: [Qin et al. \(2020\)](#)

família YOLO, se consolidam como uma ferramenta essencial para a detecção de objetos em tempo real devido a sua confiável precisão e velocidade, sendo amplamente utilizada em robótica, veículos autônomos, sistemas de vigilância por vídeo, como também na esfera da agricultura para detectar e classificar culturas, pragas e doenças.

O modelo **YOLO** e suas famílias, desenvolvido por [Redmon \(2016\)](#), foi introduzido inicialmente na Conferência de Visão computacional e Reconhecimento de Padrões de 2016 como uma abordagem pioneira de detecção de objetos em tempo real de ponta a ponta. O nome “*You Only Look Once*”, você só olha uma vez, traduzindo para o português, reflete sua principal inovação que é realizar a detecção com apenas uma única passagem pela rede. Isso o diferenciou de métodos anteriores, que dependiam de janelas deslizantes, também conhecida como *sliding window*, combinadas com classificadores executados inúmeras vezes por imagem, aumentando tempo de processamento que em domínios como robótica ou segurança são muito importantes, ou de técnicas mais recentes que dividiam o processo em duas etapas, começando com a identificação de regiões propostas e, em seguida, classificação da região. Os modelos YOLO simplificaram esse fluxo ao adotar uma saída direta baseada em regressão para prever as detecções.

Neste trabalho foi utilizado o modelo YOLOv8 (*You Only Look Once*, versão 8) que vem com novas funcionalidades em relação às versões anteriores, como apresentado na [Tabela 1](#).

Tabela 1 – Versões dos modelos YOLO e suas contribuições.

Versão	Data	Contribuição	Framework
<b>V1</b>	2015	Detector de objetos de disparo único	Darknet
<b>V2</b>	2016	Treinamento multi-escala, agrupamento dimensional	Darknet
<b>V3</b>	2018	Bloco SPP, Darknet-53	Darknet
<b>V4</b>	2020	Ativação baseada em Mish, backbone CSPDarknet-53	Darknet
<b>V5</b>	2020	Detecção sem âncora, ativação baseada em SWISH, PANet	PyTorch
<b>V6</b>	2022	Autoatenção, detecção de objetos sem âncora	PyTorch
<b>V7</b>	2022	Transformadores, reparametrização E-ELAN	PyTorch
<b>V8</b>	2023	GANs, detecções sem âncora	PyTorch

Fonte: [Alif e Hussain \(2024\)](#)

### 2.3.6.1 Arquitetura YOLOv8

O modelo de visão computacional YOLOv8 é a oitava versão da série YOLO de modelos de detecção e classificação de objetos desenvolvida pela Ultralytics. Esta versão melhora as anteriores com uma arquitetura mais simplificada, melhor desempenho indo desde velocidade no processamento até melhores acurárias e possuindo mais flexibilidade.

O ponto principal do *framework* YOLOv8 é que pode ser usado para diferentes tarefas como detecção, classificação, segmentação e estimativa de pose, por essa razão deve-se considerar que para cada nível de tarefa existirá uma rede modelada especificamente para tal fim, além disso o *framework* vem com níveis de tamanhos de redes como *nano*, *small*, *medium*, *large*, *extra large* o que adiciona ou tira camadas também alterando a sua estrutura ([SOHAN; RAM; REDDY, 2024](#)).

A arquitetura do modelo YOLOv8 está dividida em 3 partes principais: *backbone*, *neck* e *head*.

#### 2.3.6.1.1 *Backbone*

*Backbone* desempenha um papel crucial na extração de informações relevantes da imagem de entrada. A extração de características realizada pelo *backbone* é feito normalmente utilizando uma [RNC](#) pré-treinada em grandes conjuntos de dados, como por exemplo, a ImageNet<sup>2</sup>. O *backbone* aprende a capturar informações e características em diferentes níveis de detalhes ou hierarquias. As primeiras camadas do *backbone* focam em padrões básicos como bordas e texturas, enquanto camadas mais profundas captam características mais complexas, como partes de objetos e informações semânticas.

O YOLOv8 utiliza uma nova rede *backbone*, que é uma versão aprimorada da arquitetura CSPDarknet53, composta por 53 camadas convolucionais. Essa arquitetura incorpora conexões parciais entre estágios, o que melhora o fluxo de informações entre os

<sup>2</sup> ImageNet é uma base de dados de imagens classificadas e pode ser acessada em <http://www.image-net.org>

diferentes níveis da rede. O *backbone* da oitava versão é formado por diversas camadas convolucionais organizadas de forma sequêncial.

Além disso, no módulo C2f o mapa de características é dividido, a primeira parte é concatenada com a outra metade depois de passar por camadas de convoluções resultando em recursos de alto nível com informações contextuais para aumentar a precisão da detecção. Já o módulo SPPF (*Spatial Pyramid Pooling Faster*), camada com poolings subsequentes e concatenações, identificado na [Figura 12](#) no final em azul mais escuro, e outras camadas convolucionais subsequentes processam os recursos em diferentes escalas, contribuindo para uma detecção mais eficiente e precisa ([SOHAN; RAM; REDDY, 2024](#)).

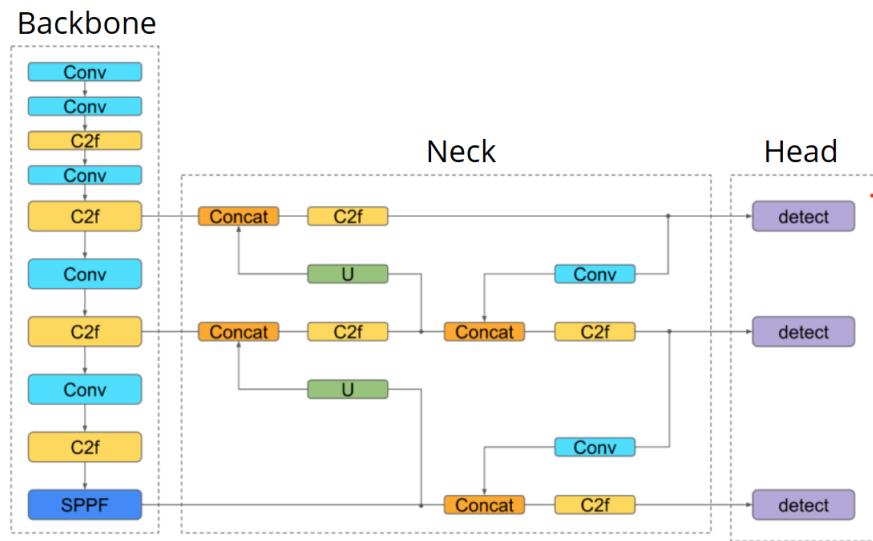


Figura 12 – Arquitetura do modelo YOLOv8 da ultralytics.

Fonte: Modificado de [Sohan, Ram e Reddy \(2024\)](#).

### 2.3.6.1.2 Neck

Segundo [Sohan, Ram e Reddy \(2024\)](#), o *neck* (pescoço) serve como uma camada intermediária que conecta o *backbone* ao estágio de previsão final. Sua função é pegar os recursos do *backbone* e refiná-los, melhorando os detalhes espaciais e a compreensão semântica. No YOLOv8, o *neck* é criado para ser destacável, o que implica que ele gerencia abordagens de maneira independente, o que torna o *framework* versátil, gerando uma estrutura para cada tipo de tarefa.

Na arquitetura do *neck* as camadas U ([Figura 12](#)), camadas que realizam a convolução transposta apresentada na [Figura 10](#), possuem objetivo de aumentar a dimensionalidade dos mapas de características, tornando possível capturar detalhes em diferentes tamanhos. O *neck* utiliza uma série de camadas convolucionais para processar esses mapas, sendo o design cuidadosamente ajustado para equilibrar velocidade e

precisão, otimizando o número de canais e os tamanhos de kernel em cada camada para maximizar o desempenho do modelo.

#### 2.3.6.1.3 *Head*

A *head* (cabeça) é a parte responsável por fazer as previsões, possuindo 3 blocos de detecção. Cada bloco lida com diferentes tipos de informações, pois recebe características de diferentes níveis das camadas do *neck*. O bloco de detecção pode ser dividido em 2 caminhos: o primeiro caminho é a detecção das caixas delimitadoras e o segundo sendo responsável pela classificação.

### 2.3.7 Métodos de amostragem

O método *holdout* é uma técnica comum em aprendizado de máquina e estatística usada para avaliar o desempenho de um modelo. O objetivo dessa abordagem é identificar a configuração de partição que produz a máxima precisão possível. As divisões mais comuns das partições está em criar um conjunto de treino, validação e teste a partir do banco de imagens (BAMI; BEHNAMPOUR; DOOSTI, 2025). Para este trabalho o conjunto de treino resultou em um total de 80% do banco, enquanto o conjunto de validação e teste ficaram com 10% das imagens cada um. Definem-se as funções de cada conjunto como:

- **Conjunto de treino:** Usado pelo modelo para aprendizado de padrões, ajustes de pesos e parâmetros;
- **Conjunto de validação:** Usado para ajustar hiperparâmetros, como por exemplo, taxa de aprendizado, número de camadas e regularização. A também validação ajuda a monitorar o desempenho do modelo durante o treinamento e previne o *overfitting* (o modelo se ajusta muito bem aos dados de treino, mas não generaliza bem para novos dados, resultando em baixo desempenho em testes), pois ocorre a avaliação do desempenho generalizado do modelo. Por último, a validação pode avisar o treinamento caso necessite uma parada antecipada;
- **Conjunto de testes:** Serve para uma avaliação sem viés do modelo, pois possui dados que não foram utilizados no treinamento.

### 2.3.8 Métricas de avaliação

Um problema com apenas duas categorias a serem diferenciadas é chamado de problema binário, no qual as classes são classificadas como positiva e negativa. Nesse contexto, define-se que:

- **Verdadeiro Positivo (VP):** ocorre quando uma amostra da classe positiva é corretamente classificada como tal.
- **Falso Positivo (FP):** ocorre quando uma amostra da classe negativa é incorretamente classificada como sendo da classe positiva.
- **Verdadeiro Negativo (VN):** corresponde à classificação correta de uma amostra pertencente à classe negativa.
- **Falso Negativo (FN):** ocorre quando uma amostra da classe positiva é erroneamente classificada como pertencente à classe negativa.

Essa avaliação pode ser representada também como uma matriz de confusão ([Tabela 2](#)):

Tabela 2 – Representação da matriz de confusão para um problema de classe binária.

		Classe Prevista	
		Positivo	Negativo
Classe Atual	Positivo	Verdadeiro Positivo	Falso Negativo
	Negativo	Falso positivo	Verdadeiro Negativo

Para a avaliação do modelo YOLOv8 para classificação de folhas de café foi utilizado a métrica Acurácia. Como explicado por [Bami, Behnampour e Doosti \(2025\)](#), a acurácia é calculada como o número de eventos previstos dividido pelo número total de eventos. Portanto a precisão é a porcentagem das previsões da classe em relação a classe total. A expressão da métrica de acurácia está descrita na Equação 2.1:

$$Acurácia(Acc) = \frac{VP + VN}{VP + VN + FP + FN}. \quad (2.1)$$

# 3 Desenvolvimento

O sistema proposto para classificação de folhas de café é composto pelas etapas de pré-processamento, extração de características e sua classificação seguido da análise e interpretação dos resultados. Para o pré-processamento ocorre a seleção de imagens a olho nu de folhas de café, tais imagens são selecionadas e classificadas em relação aos efeitos colaterais relacionados à falta de macronutrientes. A próxima etapa é a extração de características e classificação, onde o modelo YOLOv8 foi treinado com o banco de imagens para análise e interpretação dos resultados.

O sistema foi desenvolvido e treinado utilizando a linguagem Python e um computador com placa de vídeo RTX 3060, processador de 3,6 GHz (Intel Core I7-9700K) e 16 GB de memória RAM.

## 3.1 Materiais e Métodos

### 3.1.1 Banco de imagens

Em uma parceira entre o Prof. Thiago Pirola Ribeiro e Prof. Enio Tarso de Souza Costa foi realizado um experimento com mudas de café para gerar estresse nas plantas por falta de macronutrientes e, assim, conseguir demonstrar tanto visualmente quanto por meio de análises laboratoriais o que estava indicado na literatura da área.

O experimento utilizou mudas de café Arábica que, inicialmente, foram cultivadas em bandejas compartilhadas contendo uma solução nutricional completa, ou seja, foram oferecidos todos os macronutrientes essenciais em quantidades ideais para o bom desenvolvimento das plantas. Trabalhou-se com 50 mudas no estágio inicial. Após 15 dias, as plantas foram divididas em sete grupos experimentais, sendo cada grupo composto por 4 vasos (repetições) para garantir maior controle do experimento. Para seis dos grupos, retirou-se apenas um macronutriente essencial específico: cálcio (Ca), potássio (K), magnésio (Mg), nitrogênio (N), fósforo (P) e enxofre (S). O sétimo grupo permaneceu com a solução completa, contendo todos os nutrientes necessários para o desenvolvimento ideal ([Figura 13](#)).

O experimento foi conduzido durante 140 dias, no qual foram capturadas imagens em alta resolução com uma câmera Nikon D3100, configurada com ISO 400, lente de 55mm, abertura f/7.1 e velocidade de obturador de 1/60s. Ao final do período, foi criada uma base de imagens denominada BDICAFE, contendo 16.966 imagens classificadas de acordo com as deficiências nutricionais obtidas no experimento.

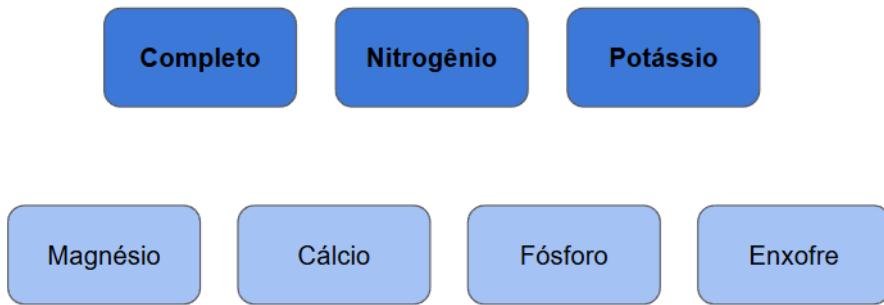


Figura 13 – Todas as classes definidas no BDICAFE. As classes completo, nitrogênio e potássio, em destaque, foram as retiradas conforme os experimentos.

### 3.1.2 YOLOv8

Neste trabalho foi utilizado o modelo YOLOv8 que conta com novas funcionalidades em relação às versões passadas ([Tabela 1](#)) em sua versão nano, que possui uma arquitetura mais leve, menos parâmetros, camadas convolucionais mais simples e por isso utiliza menos poder computacional levando a uma inferência mais rápida em tempo real. A escolha do *framework* YOLOv8 se deve principalmente a sua alta acurácia acompanhada pelo processamento em tempo real. O treinamento foi realizado utilizando a versão *nano* do modelo pré-treinado. Mais informações sobre os parâmetros estão no [Apêndice A](#).

Para a realização dos experimentos foi utilizado o dataset BDICAFE com as imagens selecionadas dos últimos 40 dias, buscando realçar os efeitos da falta de macronutrientes na planta. E como alternativas também foram adicionados filtros com aumento de contraste e remoção de fundo.

Isso se deve porque as deficiências mais comum de falta de macronutrientes aparece como a clorose nas folhas, portanto, o aumento de contraste nas imagens busca realçar essa característica.

Como última mudança no Dataset utilizou-se a remoção de fundo, sendo esta aplicada buscando retirar das imagens alguns objetos como mãos, galhos e folhas que não eram o foco principal, que poderiam atrapalhar o modelo nas suas predições.

### 3.1.3 Remoção de Fundo

Ao analisar as imagens do Dataset surgiram alguns questionamentos, pois nas imagens existem um “plano de fundo”: alguns objetos como caule, pecíolo, o vaso e até mãos poderiam impactar nos resultados. Com isso, levantou-se se a possibilidade de remoção do plano de fundo para que somente objeto (nesse caso a folha em primeiro plano) ficasse na imagem.

Neste trabalho utilizou-se do módulo `rembg` em Python. Esta ferramenta foi construída e otimizada especialmente para a remoção automática de fundo de imagens. O módulo `rembg` é construído sobre modelos de aprendizado de máquina projetados para distinguir entre o primeiro plano (objeto de interesse) e o fundo, utilizando-se de um modelo baseado em U<sup>2</sup>-Net otimizado para segmentação humana ou de objetos (Figura 11). Uma das razões do módulo ser utilizado neste trabalho é devido a praticidade, uma vez que toda a rede é abstrata, sendo somente necessário fornecer a imagem de entrada e obtém-se como resultado a mesma imagem com o fundo removido (Figura 14). Visualmente o módulo `rembg` se mostrou promissor, conseguindo retirar as informações indesejadas como galhos e mãos por exemplo, porém para uma análise mais detalhada é necessário realizar uma comparação com uma retirada manual que poderá ser realizada em um trabalho futuro.



**A - Imagem sem remoção de fundo**



**B - Imagem com o fundo removido**

Figura 14 – Exemplo de um resultado da aplicação da biblioteca `rembg` para uma imagem do Dataset BDICAFE.

### 3.1.4 Contraste

Outro questionamento que surgiu foi devido à algumas características das classes do Dataset. Pôde-se notar que as folhas com deficiências em potássio, nitrogênio e enxofre (Figura 1) possuem uma característica em comum: a clorose geralmente uniforme. Tal característica poderia afetar a separação das classes devido à suas similaridades. Com isso, optou-se pela aplicação de um filtro de contraste onde foi realçado principalmente as características de clorose devido ao aumento das diferenças das áreas claras e escuras (explicado na subseção 2.2.2). Para este trabalho foi utilizada a biblioteca `Opencv` para

aplicação do filtro: o primeiro passo foi transformar a imagem RGB para LAB, feito isso foi selecionado o canal de luz L onde foi aplicado um limite de 2 em um *grid* de 8x8. O passo final foi retornar a imagem de LAB para RGB, pois o RGB é o padrão mais comum utilizado. O resultado do contraste em uma imagem de folha de café com clorose pode ser visualizado na [Figura 15](#). Nota-se o realce das características amarelas da clorose nas imagens.



**A - Imagem sem contraste**



**B - Imagem com aumento de contraste**

Figura 15 – Exemplo do filtro de contraste aplicado sobre uma imagem do Dataset BDICAFE.

## 3.2 Experimentos

Para a obtenção das métricas dos modelos treinados, foram realizadas validações com o conjunto de teste, tal conjunto foi separado previamente, para que o modelo fosse avaliado com imagens ignoradas no treinamento e na escolha de hiperparâmetros, buscando simular uma situação mais real.

Depois de realizado a preparação dos dados foi feito o treinamento do modelo YOLOv8 com os parâmetros descritos no [Apêndice A](#) sobre o conjunto inteiro do Dataset BDICAFE, totalizando 16.966 imagens. O intuito do teste foi analisar se o modelo consegue capturar os efeitos colaterais, ou informações relevantes das faltas de macronutrientes nas plantas que são mais complexas de se analisar devido a semelhança nos estágios iniciais. Como resultado o *loss* de treino, que é uma métrica utilizada para verificar o quão bem o modelo está saindo em relação aos dados de entrada, teve uma

queda contínua enquanto o de validação apresentou alguns picos no final se mantendo em queda, foi obtido uma acurácia de 55% (Figura 16) no conjunto de testes, demonstrando que as características visuais das plantas por serem muito parecidas nos estágios iniciais, tornando difícil a previsão das classes como ilustra a Figura 17.

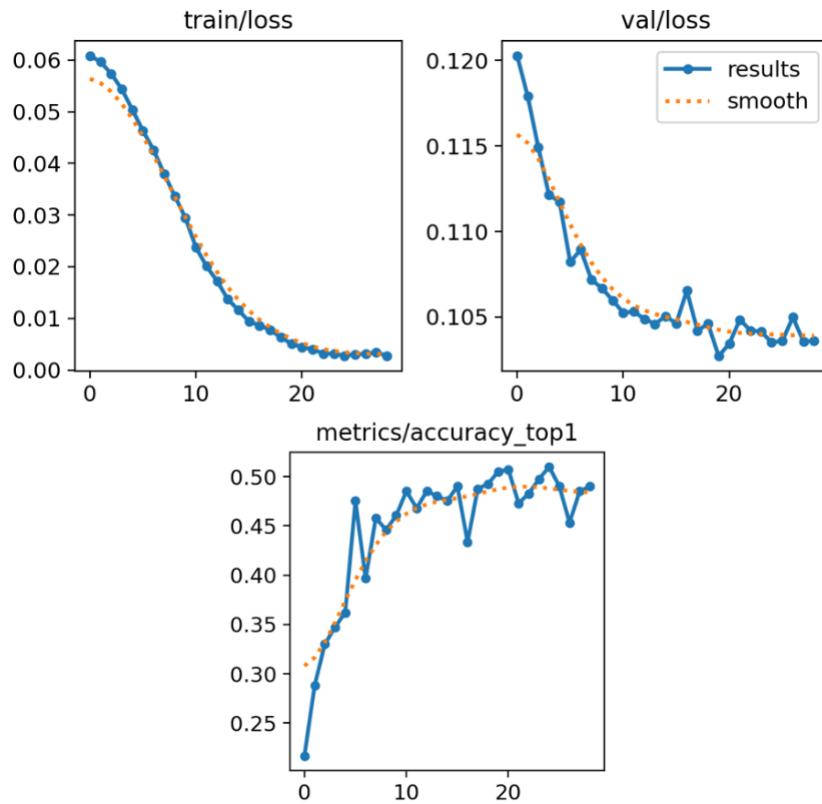


Figura 16 – Resultados do treinamento utilizando todas as imagens do Dataset BDICAFE.

Após a obtenção e análise dos resultados foi realizada uma seleção mais rigorosa das imagens contidas no Dataset. Buscou-se escolher as imagens com características mais acentuadas da falta de cada macronutriente e, para isso foram separados os últimos 40 dias do experimento, pois os efeitos visuais nas plantas já eram notados visualmente. Com isso, a partir desse momento, todos os testes foram realizados com essa nova base, ou seja, com as 669 imagens selecionadas: 96 de cálcio, 142 de completo, 76 de potássio, 95 de magnésio, 59 de nitrogênio, 102 de fósforo e 99 de enxofre. Também foi aplicada uma técnica de *data augmentation* (aumento de dados) sobre um conjunto de imagens de treino e validação composto por 600 amostras. O objetivo principal dessa etapa foi expandir a diversidade do dataset, melhorar a robustez do modelo e mitigar possíveis problemas de overfitting, dado o número limitado de imagens disponíveis inicialmente.

Utilizando os recursos nativos de *augmentation* do framework YOLOv8, o conjunto de dados foi expandido para aproximadamente o dobro de seu tamanho original, resultando em cerca de 1.200 imagens. O processo foi realizado automaticamente pelo

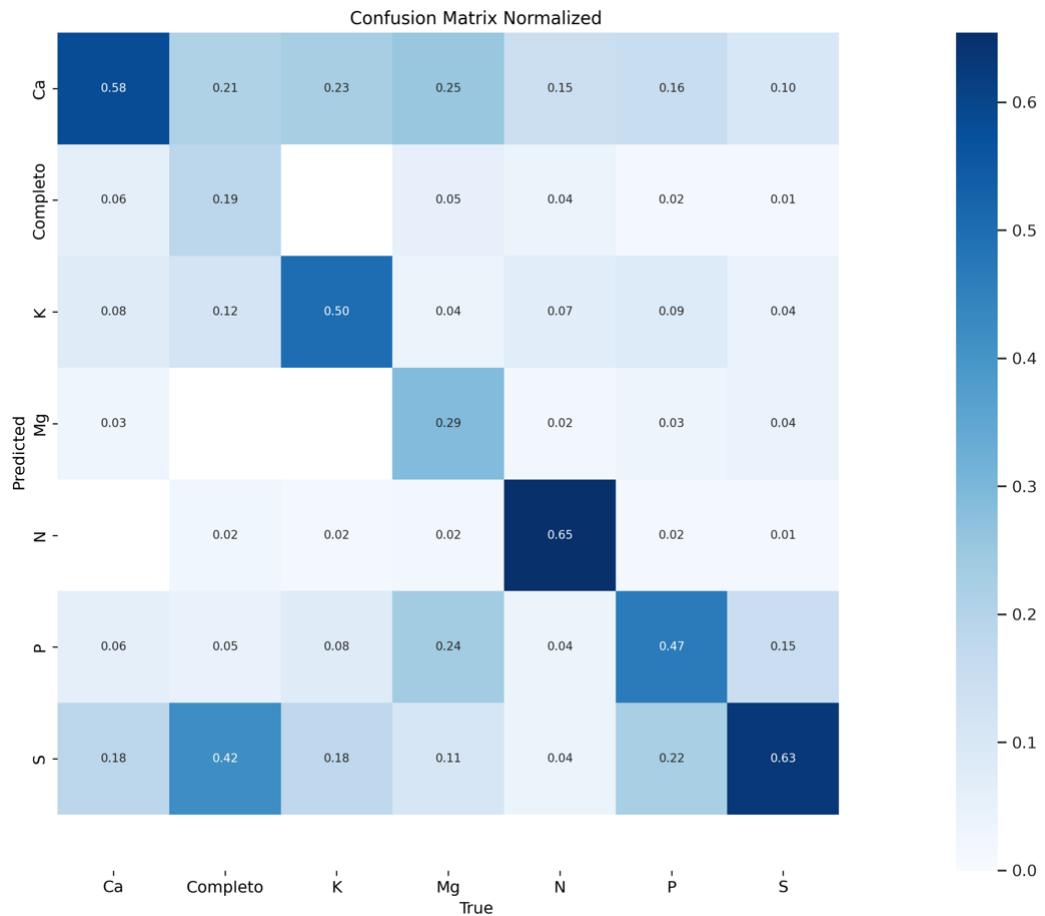


Figura 17 – Matriz de confusão dos testes realizados com todas as imagens do Dataset BDICAFE. Demonstra a dificuldade do modelo de fazer previsões, sua acurácia de 55% seria apenas uma previsão aleatória.

pipeline de treinamento da YOLOv8, que oferece uma gama de transformações aplicadas em tempo de execução.

Após a obtenção desses resultados, os testes foram divididos em três partes:

1. Sem a aplicação de filtro à imagem, ou seja, usando a imagem original;
2. Aplicando filtro para aumento de contraste na imagem original, e
3. Remoção do fundo da imagem original.

Seguiu-se a seguinte sequência para 4 os experimentos desenvolvidos, tendo para cada um a divisão baseada nas imagens:

1º: Foi treinado o modelo com todas as classes;

2º: Foi retirada a Classe Completo, devido à sua facilidade de confundir o modelo;

- 3º: Foi retirada a Classe Completo e Nitrogênio,
- 4º: Foi retirada a Classe Completo, Nitrogênio e Potássio.

O ponto principal da classe nitrogênio e potássio é que eles dividem uma característica comum de deficiência – clorose geral – podendo impactar na classificação do modelo.

### 3.2.1 1º Experimento

Nesse experimento foi utilizando o Dataset completo, ou seja, contendo todas as classes e, com essas imagens, foram realizados 3 testes: sem filtragem, aplicado aumento de contraste e, por último, realizando a remoção de fundo. As tabelas e gráficos resultantes do processamento de cada teste deste experimento também estão na [seção B.1](#) para consultas.

- **1º Experimento - Sem Filtragem**

Com o novo Dataset, ([Figura 18](#)), o *loss* de treino teve uma queda contínua enquanto o de validação apresentou um aumento no final. Acurácia aumentou, mas ainda mantendo uma instabilidade, atingindo 70,6% de acurácia no conjunto de teste. A principal importância do teste foi a matriz de confusão ([Figura 19](#)) mostrar que as principais predições erradas estão nas classes completa, ou seja, as imagens saudáveis, e nitrogênio.

Devido a dificuldade do modelo predizer as imagens saudáveis, foi retirado do Dataset essa classe com a suposição de que por algumas imagens das outras classes possuírem folhas saudáveis, ou folhas ainda no estágio inicial dos efeitos, essa semelhança poderia estar deixando o modelo indeciso.

- **1º Experimento - Aplicado Aumento de Contraste**

O modelo chegou em uma acurácia de 77,9% no conjunto de teste, o *loss* de treino teve uma queda contínua enquanto o de validação apresentou um aumento a partir da época 10, mas não o suficiente para influenciar o modelo. De modo geral a acurácia se mostrou instável com muitos picos e fundos ficando em torno de 70% ([Figura 20](#)), bem maior do que o Dataset sem filtro, mostrando que o aumento do contraste facilita a classificação do modelo. Nota-se também pela matriz de confusão ([Figura 21](#)) que a classe completo e nitrogênio influenciam no desempenho do modelo.

- **1º Experimento - Aplicado Remoção de Fundo**

O modelo chegou em uma acurácia de 65,2% no conjunto de teste, o *loss* de treino obteve uma queda contínua enquanto o da validação apresentou uma

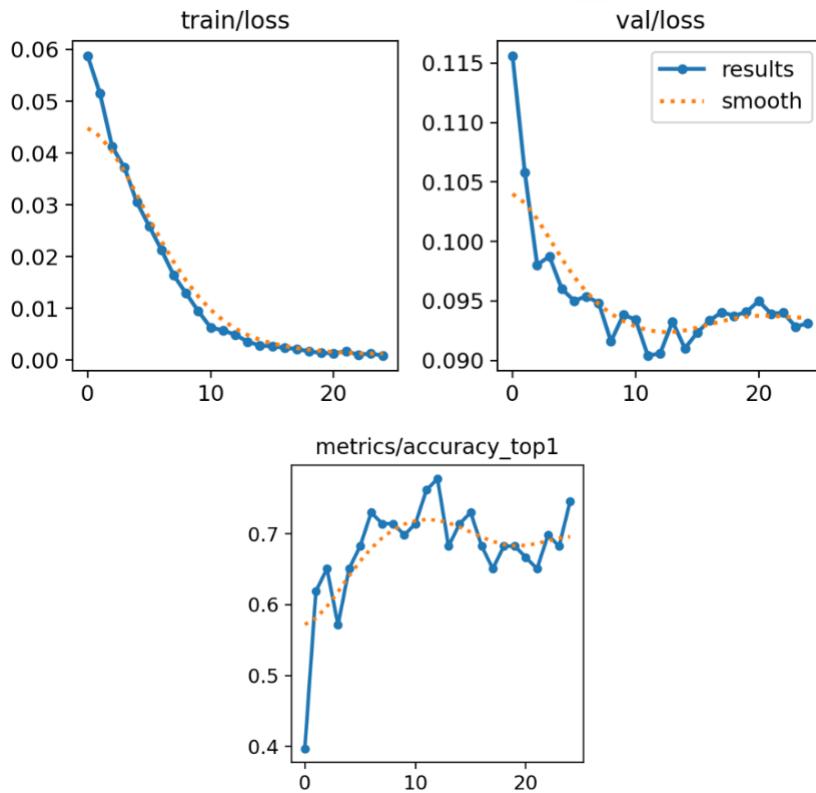


Figura 18 – Resultados do treinamento utilizando imagens selecionadas do Dataset BDICAFE.

queda no começo mas subindo para as épocas finais. A acurácia apresentou um comportamento com muitos picos e pontos que não ocorreu melhora, ficando abaixo de 70% (Figura 22), mostrando-se bem inferior ao Dataset sem filtros e contraste. Nota-se também pela matriz de confusão (Figura 23) que a classe completo e nitrogênio influenciam no desempenho do modelo.

Como é possível observar na Tabela 3, é apresentado o resumo dos valores de acurácia obtidos para o 1º experimento em que foram utilizadas todas as classes.

Tabela 3 – Tabela com a Acurácia para os testes realizados com o Dataset contendo todas as classes.

Experimento	Acurácia
Sem filtro	0,706
Com aumento de contraste	0,779
Com remoção de fundo	0,658

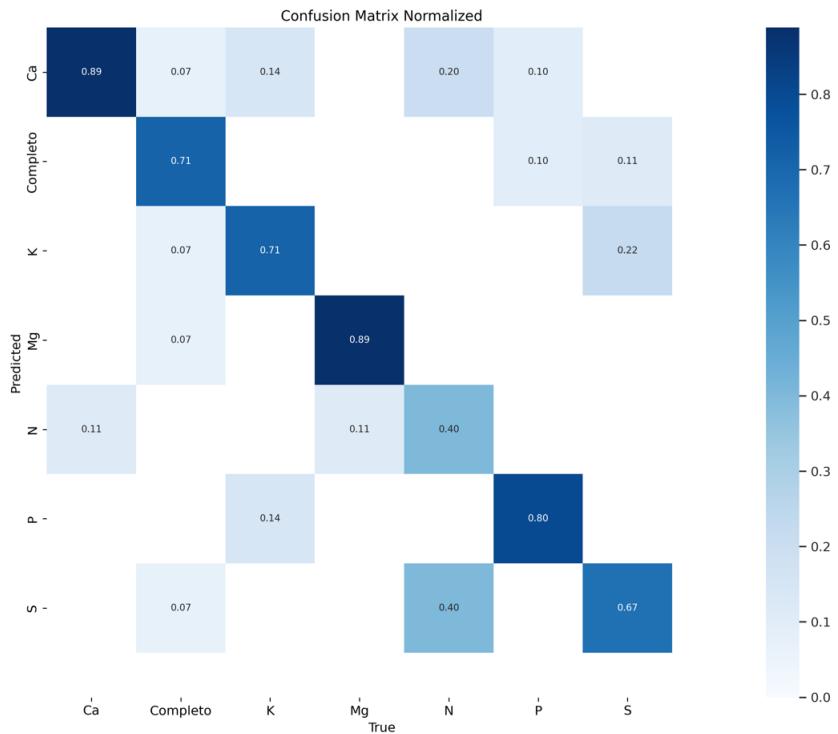


Figura 19 – Matriz de confusão dos testes realizados com as imagens selecionadas do Dataset BDICAFE.

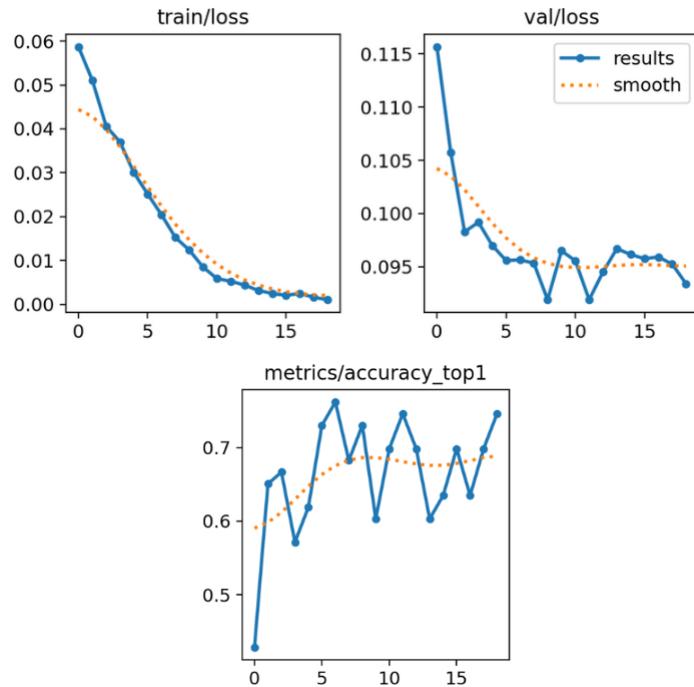


Figura 20 – Resultados do treinamento utilizando imagens selecionadas do Dataset BDICAFE aplicadas sob um filtro de aumento de contraste.

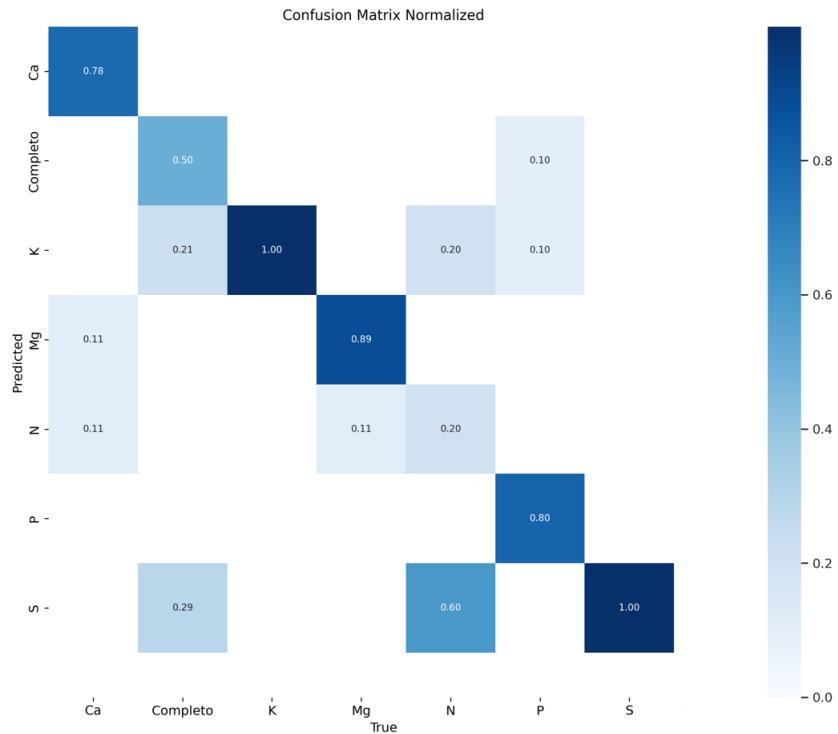


Figura 21 – Matriz de confusão dos testes realizados com as imagens selecionadas do Dataset BDICAFE aplicadas sob um filtro de aumento de contraste.

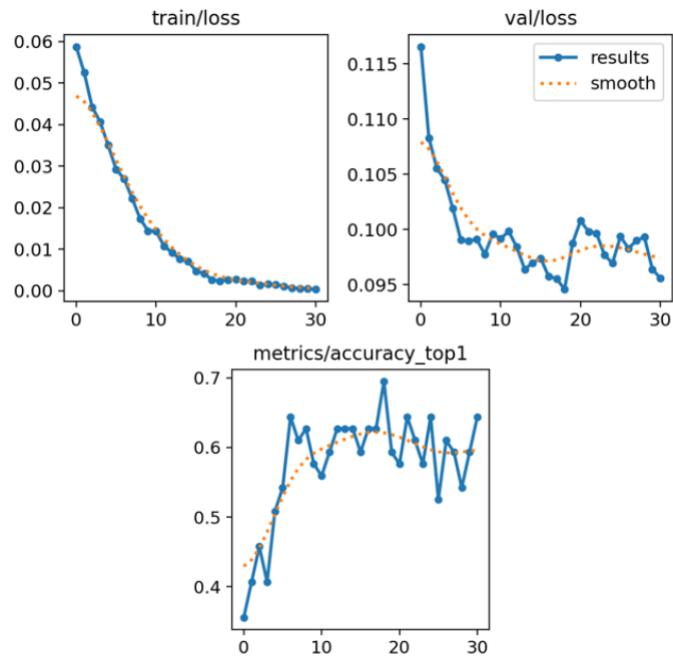


Figura 22 – Resultados do treinamento utilizando imagens selecionadas do Dataset BDICAFE, aplicadas sob um filtro de remoção de fundo.

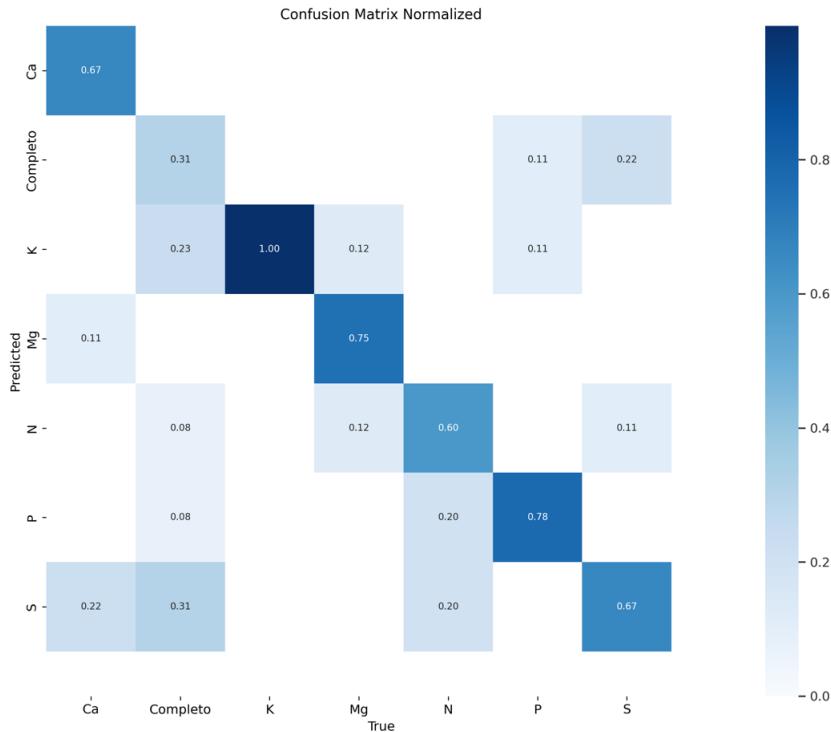


Figura 23 – Matriz de confusão dos testes realizados com as imagens selecionadas do Dataset BDICAFE, aplicadas sob um filtro de remoção de fundo.

### 3.2.2 2º Experimento

Para o segundo experimento foi retirada a classe completo, ou seja, que contém todos os nutrientes. As tabelas e gráficos resultantes do processamento de cada teste deste experimento estão na [seção B.2](#) para consultas.

- **2º Experimento - Sem a Classe Completa e Sem Filtragem**

O *loss* de treino teve uma queda contínua enquanto o de validação apresentou um aumento no final, mas diferentemente do experimento retirando a classe completa nas épocas finais o gráfico (Figura 30) apresentou uma melhora. Acurácia aumentou no começo mas piorando nas épocas finais. Foi obtido uma acurácia de 87% no conjunto de teste, mostrando assim que a suposição estava correta, a classe completa afeta de maneira significativa as outras classes, isso pode ocorrer de duas maneiras, a diferenças pouco perceptíveis em relação as outras classes, alguma folhas podem parecer saudáveis devido à uma clorose mais branda por exemplo, induzindo o sistema ao erro. O segundo ponto está relacionado com o Dataset, para uma melhor percepção do modelo o correto seria mais imagens com características notáveis de deficiência. Por fim, a análise da matriz de confusão (Figura 31) demonstra que a classe nitrogênio influencia na classificação do modelo, sendo a principal classe com predições erradas.

- **2º Experimento - Sem a Classe Completa e Aplicando o Aumento de Contraste**

O modelo atingiu uma acurácia de 77,8% no conjunto de teste, o *loss* de treino teve uma queda contínua enquanto o de validação apresentou um aumento no final, mas diferentemente do experimento retirando a classe completa nas épocas finais o gráfico ([Figura 32](#)) apresentou uma melhora. Acurácia apresentou picos mas se manteve entre 70-80%, não apresentando melhorias em comparação ao teste com todas as classes mas também sendo bem inferior ao teste sem filtro, observou-se também na matriz de confusão ([Figura 33](#)) o mesmo padrão de interferência da classe nitrogênio.

- **2º Experimento - Sem a Classe Completa e Aplicando a Remoção de Fundo**

Foi obtido acurácia de 75,5% no conjunto de teste, o *loss* de treino e validação tiveram uma queda contínua ([Figura 34](#)). A acurácia apresentou um comportamento com muitos picos, mantendo-se entre 70-80%, não apresentando melhorias em relação aos testes sem a classe completa mas atingiu melhor acurácia comparado com teste utilizando todo o Dataset. Observou-se também na matriz de confusão ([Figura 35](#)) o mesmo padrão de interferência da classe nitrogênio e na classe enxofre, seguiu-se pelo mesmo caminho retirando assim a classe nitrogênio.

Como é possível observar na [Tabela 4](#), é apresentado o resumo dos valores de acurácia obtidos para o 2º experimento em que foram utilizadas todas as classes do Dataset exceto a classe completa.

Tabela 4 – Tabela com a Acurácia para os testes realizados com o Dataset sem a classe completa.

Experimento	Acurácia
Sem filtro	0,870
Com aumento de contraste	0,778
Com remoção de fundo	0,755

### 3.2.3 3º Experimento

Para o terceiro experimento foi retirado a classe completo e nitrogênio. As tabelas e gráficos resultantes do processamento de cada teste deste experimento estão na [seção B.3](#) para consultas.

- **3º Experimento - Sem a Classe Completa e Nitrogênio e Sem Filtragem**

Como resultado foi obtido 95,8% de acurácia para o conjunto de teste sem filtro. O *loss* de treino teve uma queda contínua enquanto o de validação apresentou um aumento no final, aumentando um pouco mais nas últimas épocas. Acurácia aumentou no começo, atingindo seu pico rápido e depois se mantendo mais abaixo do pico nas últimas épocas. A ótima acurácia do modelo se deve principalmente em relação à clorose geral, efeito consequente da falta de nitrogênio. Devido a classe nitrogênio não estar no Dataset o modelo pode distinguir melhor o resto das classes, demonstrando que a clorose referente à falta de potássio e enxofre são mais distinguíveis, pode-se afirmar também que a necrose na folha e nas pontas devido à falta de potássio é uma característica a mais para facilitar a predição do modelo. Por final, analisando a matriz de confusão, o modelo se mostrou com mais dificuldades em relação à classe enxofre, mas como foram erros menos significativos optou-se pela retirada da classe potássio para o experimento da [subseção 3.2.4](#), que apresentou dificuldades maiores para detecção no teste sem a classe completa.

- **3º Experimento - Sem a Classe Completa e Nitrogênio e Aplicando o Aumento de Contraste**

O modelo atingiu uma acurácia de 89,6% no conjunto de teste, o *loss* de treino e validação tiveram uma queda contínua apresentando uma melhora. A acurácia melhorou comparado com o teste sem a classe completa apresentando uma subida nas primeiras épocas mas caindo nas proximas épocas, melhorando sua previsão em relação ao teste sem a classe completa, mostrando a influência da classe nitrogênio, observou-se também na matriz de confusão que a classe potássio possui mais previsões erradas.

- **3º Experimento - Sem a Classe Completa e Nitrogênio e Aplicando a Remoção de Fundo**

Foi obtido pelo modelo uma acurácia de 85,1% no conjunto de teste, o *loss* de treino e validação tiveram uma queda contínua e a acurácia apresentou um comportamento com muitos picos, mantendo-se perto de 80%, melhorando sua previsão em relação ao teste sem a classe completa e mostrando a influência da classe nitrogênio, mas infelizmente ficando bem abaixo dos modelos treinados usando o Dataset sem filtros e o Dataset com filtro de aumento de contraste, observou-se também na matriz de confusão que a classe potássio possui previsões erradas e seguiu-se pelo mesmo caminho dos outros testes.

Como apresentado na [Tabela 5](#), é possível observar o resumo dos valores de acurácia obtidos para o 3º experimento em que foram utilizadas todas as classes do Dataset exceto as classes completa e nitrogênio.

Tabela 5 – Tabela com a acurácia dos testes realizados com o Dataset sem a classe completa e nitrogênio

Experimento	Acurácia
<b>Sem filtro</b>	0,958
<b>Com aumento de contraste</b>	0,896
<b>Com remoção de fundo</b>	0,851

### 3.2.4 4º Experimento

Para o quarto experimento foram retiradas as classes completa, nitrogênio e potássio. As tabelas e gráficos resultantes do processamento de cada teste deste experimento estão na [seção B.4](#) para consultas.

- **4º Experimento - Sem a Classe Completa, Nitrogênio e Potássio e Sem Filtragem**

O ultimo teste realizado foi retirando a classe potássio, junto com completo e nitrogênio mas não obteve um resultado significativo ou superior, ficando assim com 95% de acurácia no conjunto de teste, o *loss* de treino teve uma queda contínua enquanto o de validação apresentou um aumento a partir da época 10, mas não o suficiente para influenciar o modelo. Acurácia aumentou já nas primeiras épocas passando de 90% de acurácia e se mantendo alta ao longo do treinamento, obteve-se uma matriz de confusão ótima, mas no final demonstrando que a retirada da classe potássio não influencia muito no modelo.

- **4º Experimento - Sem a Classe Completa, Nitrogênio e Potássio e Aplicando o Aumento de Contraste**

Como teste final aplicado sobre o filtro de aumento de contraste retirou-se a classe completo, nitrogênio e potássio, conseguindo assim 95% de acurácia no conjunto de teste, o *loss* de treino e validação tiveram uma queda contínua apresentando uma melhora . A acurácia apresentou um comportamento com muitos picos e pontos que não ocorreu melhora, mantendo-se entre 80-90%, mostrando-se inferior referente aos outros testes sem filtro e com aumento de contraste. A matriz de confusão não mostra mais interferências de outras classes.

- **4º Experimento - Sem a Classe Completa, Nitrogênio e Potássio e Aplicando a Remoção de Fundo**

Com o filtro de remoção de fundo o modelo chegou a 92,3% de acurácia no conjunto de teste, o *loss* de treino e validação tiveram uma queda contínua, com a validação sofrendo um aumento no final. A acurácia apresentou um comportamento com muitos picos, mantendo-se entre 80-90% não apresentando melhoras significativas

em relação ao contraste ou ao Dataset sem filtro mas mostrou melhora em relação aos testes com remoção de fundo. A matriz aponta predições erradas principalmente da classe fósforo.

Como apresentado na [Tabela 6](#), é possível observar o resumo dos valores de acurácia obtidos para o 3º experimento em que foram utilizadas todas as classes do Dataset exceto as classes completa, nitrogênio e potássio.

Tabela 6 – Tabela com a acurácia dos testes realizados com o Dataset sem a classe completa, nitrogênio e potássio

Experimento	Acurácia
<b>Sem filtro</b>	0,950
<b>Com aumento de contraste</b>	0,950
<b>Com remoção de fundo</b>	0,923

### 3.2.5 Análise das Acurárias dos Experimentos

Este estudo apresenta uma análise em tabelas dos experimentos realizados e foi organizado em três categorias: sem filtro, com aumento de contraste e com remoção de fundo. Para cada tabela está detalhado os resultados para cada tipo de filtro e cada seleção de classes utilizada no experimento.

#### 3.2.5.1 Acurácia do Dataset sem filtro

A acurácia geral considerando todas as classes foi de 70,6%, o que indica um desempenho ruim, mas com espaço para melhorias, possivelmente devido à alta variabilidade visual entre os tipos de deficiência.

Ao remover a classe “completo” (folhas sem deficiência), a acurácia subiu significativamente para 87%. Esse aumento sugere que a classe "completo" pode estar visivelmente próximo a alguns estágios iniciais de deficiência, não demonstrando assim tantas diferenças visuais, dificultando a distinção da rede.

Quando se exclui também a classe Nitrogênio (N), classe que apresenta sintomas que podem ser confundidos com outras deficiências, a acurácia atinge 95,8%, indicando que esta classe representa um desafio adicional para a detecção precisa.

Por fim, com a remoção das classes completo, nitrogênio e potássio, a acurácia se mantém elevada, em 95%, sendo praticamente igual ao experimento anterior, demonstrando que não há necessidades de retirar mais classes.

A acurácia dos testes utilizando o Dataset BDICAFE com imagens selecionadas pode ser vista na [Tabela 7](#).

Tabela 7 – Tabela com a Acurácia dos testes realizados com o Dataset sem filtros adicionados

Dataset	Acurácia
<b>Todas as classes</b>	0,706
<b>Sem completo</b>	0,870
<b>Sem completo e N</b>	0,958
<b>Sem completo, N e K</b>	0,950

### 3.2.5.2 Acurácia do Dataset com aumento de contraste

Quando avaliadas todas as classes, a acurácia foi de 77,9%, mostrando que, mesmo com a diversidade de sintomas, o modelo manteve um desempenho estável. Notavelmente, ao remover a classe completo, a acurácia praticamente não se alterou, sugerindo que o aumento de contraste não teve grande efeito em diferenciar folhas saudáveis das deficientes, possivelmente porque os sintomas de deficiência já eram mais evidentes visualmente.

A exclusão da classe Nitrogênio (N) resultou em uma acurácia significativamente maior de 89,6%, o que indica que a deficiência de nitrogênio ainda representa um desafio considerável para o modelo, mesmo com o contraste aumentado. Isso reforça a hipótese de que os sintomas dessa deficiência são mais discretos ou similares a outras, exigindo talvez técnicas adicionais além do contraste para melhor distinção.

Por fim, com a retirada das classes completo, Nitrogênio e potássio, a acurácia atinge 95%, demonstrando que o modelo responde muito bem ao contraste quando lida apenas com classes cujos sintomas são mais visualmente destacados. Isso reforça o benefício da aplicação do filtro de contraste como uma etapa de pré-processamento que pode aumentar a clareza das características visuais relevantes para a classificação.

Em resumo, o aumento de contraste contribuiu positivamente para a acurácia geral, especialmente ao lidar com deficiências mais visíveis, e mostrou-se uma etapa promissora no aprimoramento da detecção como demonstrado na [Tabela 8](#)

Tabela 8 – Tabela com a Acurácia dos testes realizados com o Dataset com aumento de contraste

Dataset	Acurácia
Todas as classes	0,779
Sem completo	0,778
Sem completo e N	0,896
Sem completo, N e K	0,950

### 3.2.5.3 Acurácia do Dataset com remoção de fundo

Com todas as classes presentes, a acurácia foi de 65,2%, a mais baixa entre os cenários testados até o momento. Isso sugere que, embora a remoção do fundo possa reduzir ruídos visuais, ela também pode eliminar informações contextuais úteis como bordas de folhas que o modelo poderia estar utilizando para auxiliar na classificação.

Ao remover a classe completo, a acurácia subiu para 75,5%, indicando que a distinção entre folhas saudáveis e deficientes ainda era um desafio significativo, mesmo com o fundo removido. Essa melhora mostra que, ao focar apenas nas deficiências, o modelo conseguiu se concentrar melhor nos padrões visuais relevantes.

Excluindo a completo e classe Nitrogênio, a acurácia aumentou para 85,1%, reforçando que os sintomas relacionados à falta de nitrogênio continuam sendo difíceis de identificar, mas que a remoção de fundo ajudou a melhorar a performance nesse contexto.

Por fim, ao retirar também a classe Potássio, a acurácia atinge 92,3%, mostrando um bom desempenho geral com classes visualmente mais distintas. Esse resultado evidencia que, apesar da acurácia inicial mais baixa, a remoção de fundo pode beneficiar a rede quando o foco está em deficiências com padrões claros e bem definidos, ao reduzir distrações e uniformizar a entrada visual.

A acurácia dos testes utilizando o Dataset BDICAFE com imagens selecionadas e aplicadas sob um filtro de remoção de fundo podem ser vistos na [Tabela 9](#).

Tabela 9 – Tabela com a Acurácia dos testes realizados com o Dataset com remoção de fundo

Dataset	Acurácia
<b>Todas as classes</b>	0,652
<b>Sem completo</b>	0,755
<b>Sem completo e N</b>	0,851
<b>Sem completo, N e K</b>	0,923

### 3.2.5.4 Melhores Acuráncias

A [Tabela 10](#) ilustra as melhores Acuráncias dos testes realizados anteriormente. Percebe-se que quando todas as classes foram consideradas, a aplicação do filtro de aumento de contraste resultou em uma acurácia razoável. Isso sugere que o contraste realçado ajuda a evidenciar padrões visuais importantes, mas ainda há limitações na diferenciação entre classes com sintomas semelhantes, como a classe completo e deficiências leves.

Ao remover a classe completo, mesmo sem aplicar qualquer filtro nas imagens, observou-se um aumento expressivo na acurácia. Isso indica que a presença de folhas

Tabela 10 – Tabela com as melhores Acurárias dos testes realizados.

Dataset	Acurácia	Experimento
Todas as classes	0,779	Aumento de Contraste
Sem completo	0,870	Sem filtragem
Sem completo e N	0,958	Sem filtragem
Sem completo, N e K	0,950	Sem filtragem e Aumento de Contraste

saudáveis gera ambiguidade na classificação, e que o modelo é mais eficaz ao lidar apenas com folhas que apresentam algum sintoma de deficiência.

A exclusão adicional da classe Nitrogênio, cujos sintomas podem ser mais sutis ou facilmente confundidos com outras deficiências, levou a um salto considerável na acurácia. Esse valor reflete a alta capacidade do modelo de distinguir deficiências mais visíveis, sugerindo que a falta de nitrogênio é uma das classes mais desafiadoras para a rede.

Mesmo com as classes completo, nitrogênio e potássio removidas a acurácia se manteve a mesma dos experimento retirando somente a classe completo e nitrogênio, o mesmo ocorreu com a aplicação do aumento de contraste. Isso demonstra que independente da filtragem, ou seja, sem filtrou ou com aumento de contraste, não ocorreram melhorias podendo dizer que piorou já que logicamente retirando um problema visual deveria ocorrer aumento de acurácia, o que não foi o caso.

## 4 Conclusão

Neste Trabalho, estudou-se a eficiência do uso do framework YOLOv8 para classificação de deficiência de macronutrientes nas folhas de café. Também foram analisadas algumas abordagens de aplicações de filtros de contraste e remoção de fundo buscando formas de ajudar o modelo a conseguir uma melhor predição.

Os resultados demonstram que o modelo YOLOv8 consegue extrair características suficientes em um dataset complexo com poucas imagens e classes desbalanceadas, conseguindo atingir 95,8% de acurácia. As principais classes que influenciaram nos erros do modelo foram a classe completa e a classe nitrogênio. A classe completa influencia principalmente na parte referente as características comuns das folhas, muitas folhas ainda estão nos estágios iniciais dos efeitos podendo confundir a classificação, o que leva a outro ponto a ser analisado, o dataset poderia ter mais imagens de deficiências em estágios finais. Assim, o modelo poderia alcançar acurácia ainda mais altas sem necessidade de retirar classes. Em relação a classe nitrogênio, a característica principal da falta desse macronutriente leva a planta a um estado de clorose, onde a folha fica amarelada, e que também é uma característica comum na falta de enxofre e de potássio demonstrando que o modelo tem mais dificuldade para separar as classes com tantos efeitos em comum.

É importante destacar que o filtro de aumento de contraste não ajudou a melhorar o modelo, apesar que nos primeiros testes incluindo todas as classes o contraste demonstrou realçar os efeitos nas plantas facilitando a classificação. De todo modo, quanto mais classes conseguirem ser analisadas pelo modelo traz mais benefícios para o agricultor, levando a crer que não é necessário a aplicação dos filtros, pois foram retiradas mais classes para obter o mesmo resultado dos testes sem filtro, isso se deve pois o modelo pode-se ajustar em relação a essa complexidade.

Em relação aos resultados obtidos pelo uso do filtro de remoção de fundo, a característica principal é a extrema exatidão que o filtro é aplicado, separando muito bem o objeto do fundo, mas infelizmente o modelo YOLOv8 não apresentou melhorias significativa comparadas com as imagens normais, chegando a conclusão que o próprio modelo consegue separar os objetos durante seu aprendizado. Portanto é de se sugerir o estudo de remoção de fundo com redes neurais mais complexas que a YOLO, podendo levar assim a resultados melhores.

## 4.1 Trabalhos Futuros

Para trabalhos futuros, pretende-se investigar versões mais recentes da família YOLO, uma vez que novos frameworks costumam trazer melhorias significativas no processamento das imagens, resultando em classificações mais rápidas e robustas. Essas atualizações tecnológicas também podem ajudar a minimizar problemas de *overfitting*, elevando o potencial de alcançar resultados ainda mais promissores na detecção de deficiências nutricionais.

Convém destacar ainda que, embora o filtro de remoção de fundo não tenha se revelado vantajoso no presente trabalho — em parte devido ao aumento de tempo gerado pela complexidade da rede —, tal técnica pode ser aproveitada em outros modelos de Redes Neurais Convolucionais (RNCs) mais robustos, como também ser realizado uma análise mais detalhada comparando com a retirada manual. Como o método isola apenas o objeto de análise, ele se mostra muito efetivo para problemas em que o tempo de resposta não seja o principal fator, podendo justificar a adoção de arquiteturas maiores e mais exigentes em termos computacionais. Outro ponto importante a ser estudado é o teste de novos parâmetros para o filtro contraste, pois o realce da clorose pode ser um dos fatores auxiliadores ao modelo.

Por último, pretende-se analisar outras métricas além da acurácia, como recall, precision e F1-score. Essas medidas fornecem uma avaliação mais detalhada do desempenho em cada classe de deficiência e ajudam a identificar pontos específicos de melhoria no processo de classificação.

# Referências

ALBAWI, S.; MOHAMMED, T. A.; AL-ZAWI, S. Understanding of a convolutional neural network. In: **IEEE. 2017 international conference on engineering and technology (ICET)**. Antalya, Turkey, 2017. p. 1–6. Citado na página [21](#).

ALIF, M. A. R.; HUSSAIN, M. Yolov1 to yolov10: A comprehensive review of yolo variants and their application in the agricultural domain. **arXiv preprint arXiv:2406.10139**, 2024. Citado 3 vezes nas páginas [14](#), [21](#) e [29](#).

BAMI, Z.; BEHNAMPOUR, A.; DOOSTI, H. A new flexible train-test split algorithm, an approach for choosing among the hold-out, k-fold cross-validation, and hold-out iteration. **arXiv preprint arXiv:2501.06492**, 2025. Citado 2 vezes nas páginas [31](#) e [32](#).

CHALFOUN, S.; REIS, P. História da cafeicultura no brasil. **Café arábica: do**, 2010. Citado na página [13](#).

FAQUIN, V. Diagnose do estado nutricional das plantas. **Lavras: UFLA/FAEPE**, v. 1, p. 77, 2002. Citado na página [16](#).

GONZALEZ, R. **Digital Image Processing**. New Jersey: Pearson Education, 2009. ISBN 9788131726952. Citado na página [19](#).

HUNGILO, G. G.; EMMANUEL, G.; EMMANUEL, A. W. R. Image processing techniques for detecting and classification of plant disease: A review. In: **Proceedings of the 2019 International Conference on Intelligent Medicine and Image Processing**. New York, NY, USA: Association for Computing Machinery, 2019. (IMIP '19), p. 48–52. ISBN 9781450362696. Citado na página [18](#).

JIANG, W.; LI, Z. Automated high-precision recognition of solar filaments based on an improved u2-net. **Universe**, MDPI, v. 10, n. 10, p. 381, 2024. Citado na página [27](#).

LOESDAU, M.; CHABRIER, S.; GABILLON, A. Hue and saturation in the rgb color space. In: **SPRINGER. International conference on image and signal processing**. Cham: Springer International Publishing, 2014. p. 203–212. Citado na página [18](#).

MA, W.; LU, J. An equivalence of fully connected layer and convolutional layer. **arXiv preprint arXiv:1712.01252**, 2017. Citado na página [26](#).

QIN, X.; ZHANG, Z.; HUANG, C.; DEHGHAN, M.; ZAIANE, O. R.; JAGERSAND, M. U2-net: Going deeper with nested u-structure for salient object detection. **Pattern recognition**, Elsevier, v. 106, p. 107404, 2020. Citado 2 vezes nas páginas [27](#) e [28](#).

REDMON, J. You only look once: Unified, real-time object detection. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. Los Alamitos, CA, USA: IEEE Computer Society, 2016. Citado 2 vezes nas páginas [14](#) e [28](#).

SCHWARZ, M. W.; COWAN, W. B.; BEATTY, J. C. An experimental comparison of rgb, yiq, lab, hsv, and opponent color models. **ACM Transactions on Graphics (tog)**, ACM New York, NY, USA, v. 6, n. 2, p. 123–158, 1987. Citado na página [19](#).

SIDDIQUE, N.; PAHEDING, S.; ELKIN, C. P.; DEVABHAKTUNI, V. U-net and its variants for medical image segmentation: A review of theory and applications. **IEEE access**, IEEE, v. 9, p. 82031–82057, 2021. Citado 2 vezes nas páginas [25](#) e [26](#).

SMITH, A. R. Color gamut transform pairs. **ACM Siggraph Computer Graphics**, ACM New York, NY, USA, v. 12, n. 3, p. 12–19, 1978. Citado na página [19](#).

SOHAN, M.; RAM, T. S.; REDDY, C. V. R. A review on yolov8 and its advancements. In: JACOB, I. J.; PIRAMUTHU, S.; FALKOWSKI-GILSKI, P. (Ed.). **Data Intelligence and Cognitive Informatics**. Singapore: Springer Nature Singapore, 2024. p. 529–545. Citado 2 vezes nas páginas [29](#) e [30](#).

TERVEN, J.; CÓRDOVA-ESPARZA, D.-M.; ROMERO-GONZÁLEZ, J.-A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. **Machine Learning and Knowledge Extraction**, MDPI, v. 5, n. 4, p. 1680–1716, 2023. Citado na página [27](#).

WU, J. Introduction to convolutional neural networks. **National Key Lab for Novel Software Technology. Nanjing University. China**, v. 5, n. 23, p. 495, 2017. Citado na página [22](#).

YAMASHITA, R.; NISHIO, M.; DO, R. K. G.; TOGASHI, K. Convolutional neural networks: an overview and application in radiology. **Insights into imaging**, Springer, v. 9, p. 611–629, 2018. Citado 4 vezes nas páginas [21](#), [22](#), [24](#) e [25](#).

# Apêndices

# APÊNDICE A – Parâmetros da YOLOv8

Tabela com os respectivos parâmetros para o treinamento da YOLOv8 nano com o banco de imagens BDICAFE. Para mais detalhes verifique o [Capítulo 3](#).

Parâmetros	Valores
epoch	200
patience	6
batch	2
imgsz	640
pretrained	true
val	true
degrees	5
translate	0.1
hsv_h	0.2
hsv_s	0.7
hsv_v	0.5
fliplr	0.2
flipud	0.2
shear	0.1
scale	0.1

# APÊNDICE B – Tabelas e Gráficos dos Experimentos

Neste Apêndice estão as tabelas e gráficos dos experimentos 1, 2, 3 e 4.

## B.1 1º Experimento

O primeiro experimento foi realizado com todas as classes.

1. Dataset com todas as classes (Gráficos de Perda e Acurácia - [Figura 24](#) e Matriz de Confusão - [Figura 25](#)).

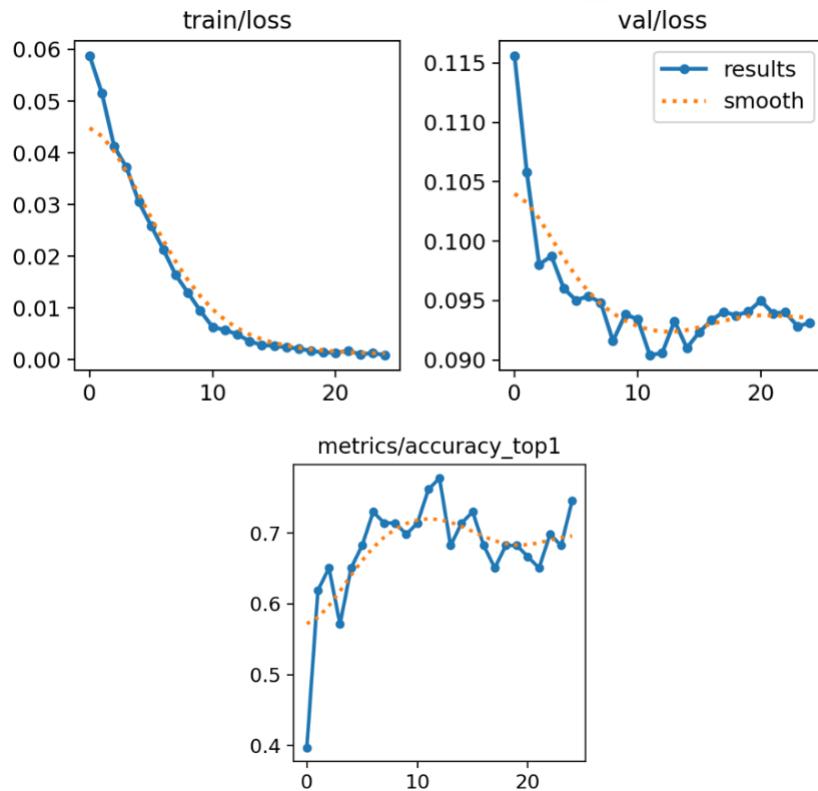


Figura 24 – Resultados do treinamento utilizando imagens selecionadas do Dataset BDICAFE.

2. Dataset com todas as classes aplicando aumento de contraste (Gráficos de Perda e Acurácia - [Figura 26](#) e Matriz de Confusão - [Figura 27](#)).
3. Dataset com todas as classes aplicando remoção de fundo (Gráficos de Perda e Acurácia - [Figura 28](#) e Matriz de Confusão - [Figura 29](#)).

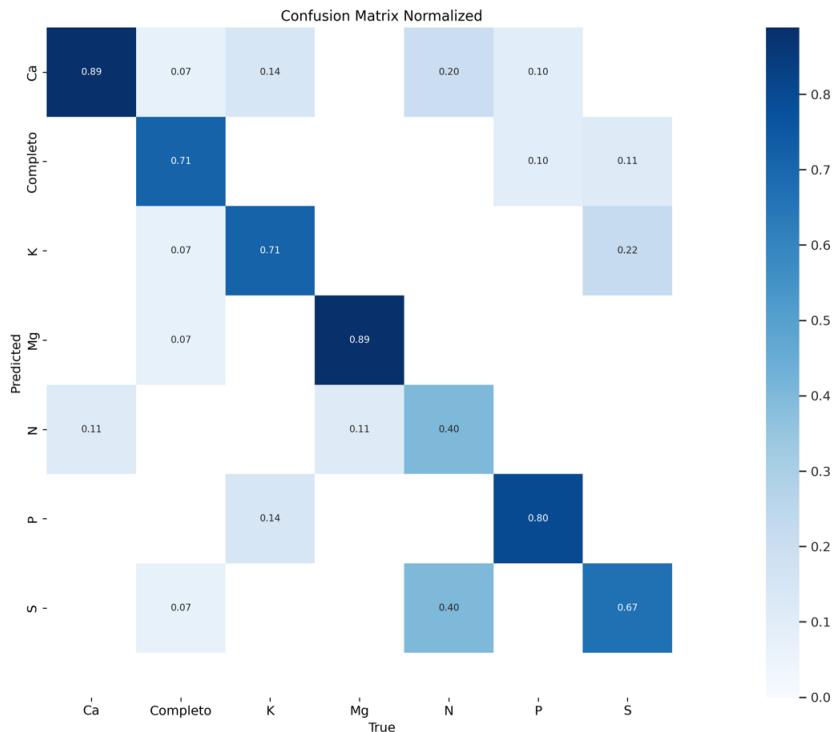


Figura 25 – Matriz de confusão dos testes realizados com as imagens selecionadas do Dataset BDICAFE.

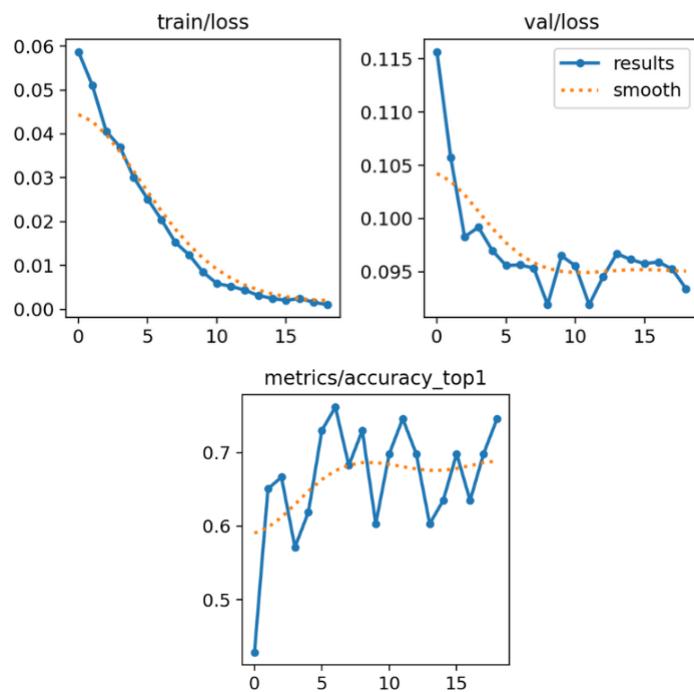


Figura 26 – Resultados do treinamento utilizando imagens selecionadas do Dataset BDICAFE aplicadas sob um filtro de aumento de contraste.

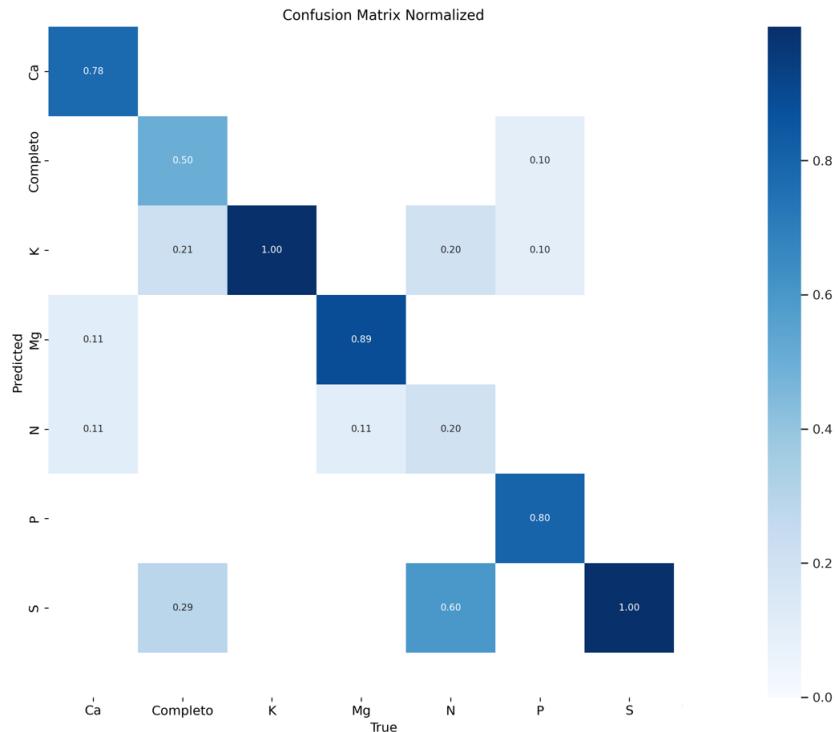


Figura 27 – Matriz de confusão dos testes realizados com as imagens selecionadas do Dataset BDICAFE aplicadas sob um filtro de aumento de contraste.

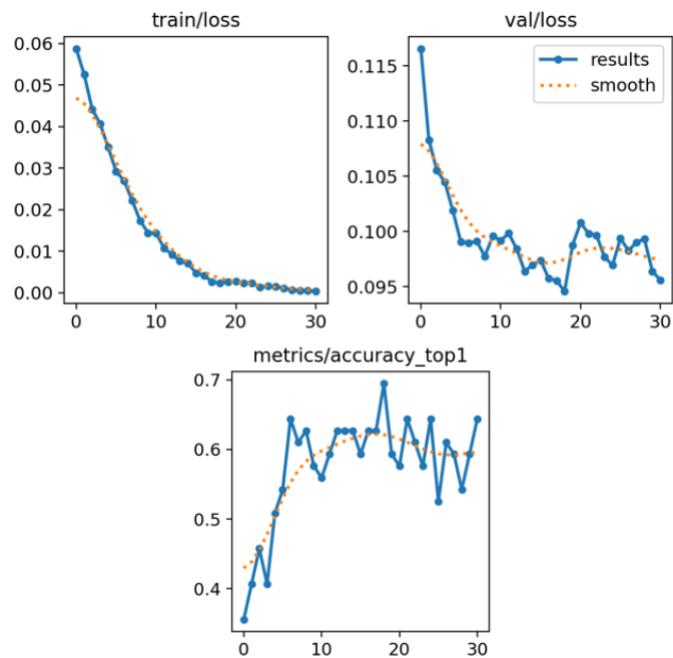


Figura 28 – Resultados do treinamento utilizando imagens selecionadas do Dataset BDICAFE, aplicadas sob um filtro de remoção de fundo.

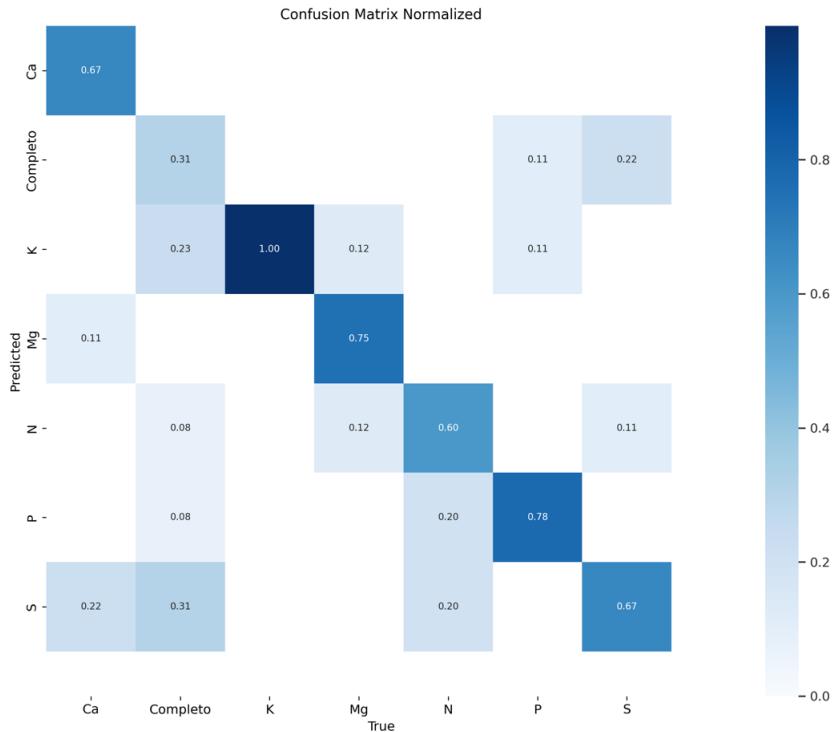


Figura 29 – Matriz de confusão dos testes realizados com as imagens selecionadas do Dataset BDICAFE, aplicadas sob um filtro de remoção de fundo.

## B.2 2º Experimento

Para o segundo experimento foi retirada a classe completo, ou seja, que contém todos os nutrientes.

1. Dataset sem a classe completa (Gráficos de Perda e Acurácia - [Figura 30](#) e Matriz de Confusão - [Figura 31](#)).
2. Dataset sem a classe completa aplicando o aumento de contraste (Gráficos de Perda e Acurácia - [Figura 32](#) e Matriz de Confusão - [Figura 33](#))
3. Dataset sem a classe completa aplicando a remoção de fundo (Gráficos de Perda e Acurácia - [Figura 34](#) e Matriz de Confusão - [Figura 35](#))

## B.3 3º Experimento

Experimentos sem a classe completo e nitrogênio

1. Dataset sem a classe completa (Gráficos de Perda e Acurácia - [Figura 36](#) e Matriz de Confusão - [Figura 37](#)).

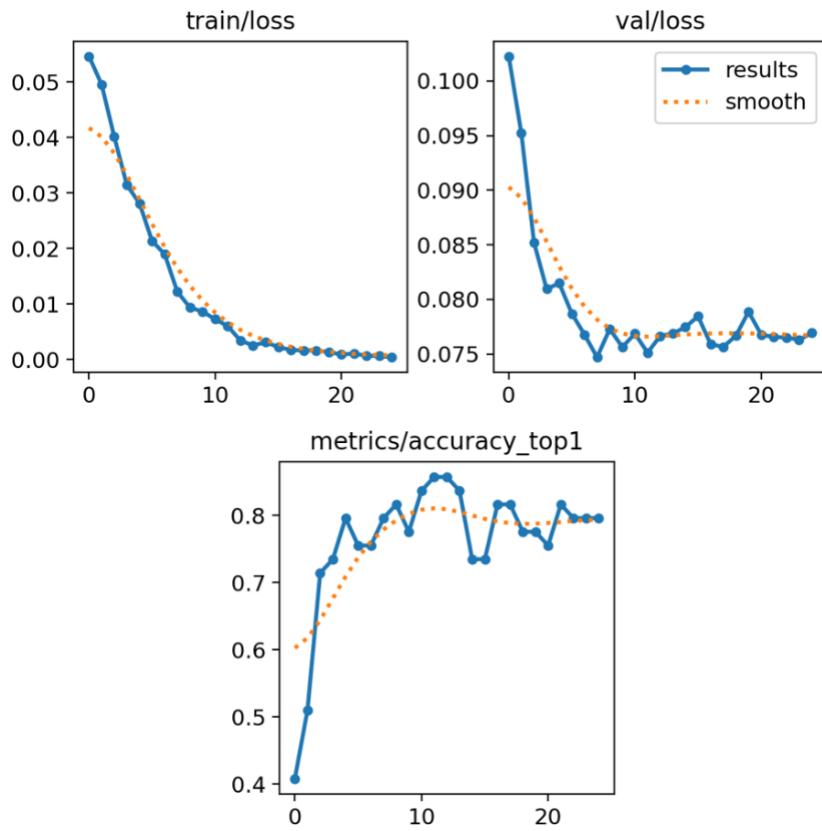


Figura 30 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE com exclusão da classe completo.

2. Dataset sem a classe completa aplicado a um filtro com aumento de contraste (Gráficos de Perda e Acurácia - [Figura 38](#) e Matriz de Confusão - [Figura 39](#)).
3. Dataset sem a classe completa aplicado a um filtro de remoção de fundo (Gráficos de Perda e Acurácia - [Figura 40](#) e Matriz de Confusão - [Figura 41](#)).

## B.4 4º Experimento

Experimentos sem a classe completo, nitrogênio e potássio

1. Dataset sem a classe completa (Gráficos de Perda e Acurácia - [Figura 42](#) e Matriz de Confusão - [Figura 43](#)).
2. Dataset sem a classe completa aplicado a um filtro com aumento de contraste (Gráficos de Perda e Acurácia - [Figura 44](#) e Matriz de Confusão - [Figura 45](#)).
3. Dataset sem a classe completa aplicado a um filtro de remoção de fundo (Gráficos de Perda e Acurácia - [Figura 46](#) e Matriz de Confusão - [Figura 47](#)).

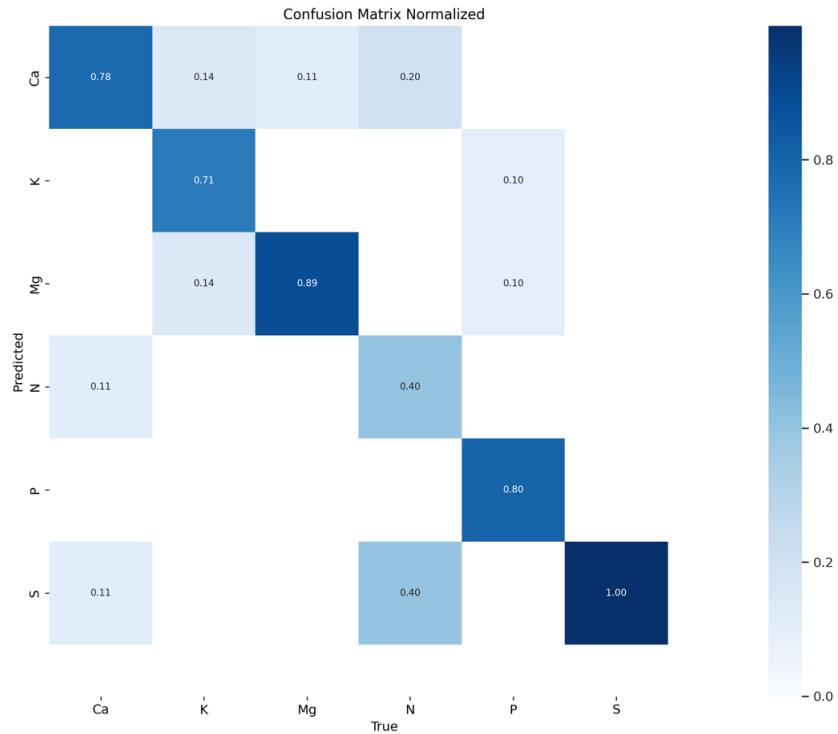


Figura 31 – Matriz de confusão dos testes realizados com as imagens selecionadas do dataset BDICAFE com exclusão da classe completo.

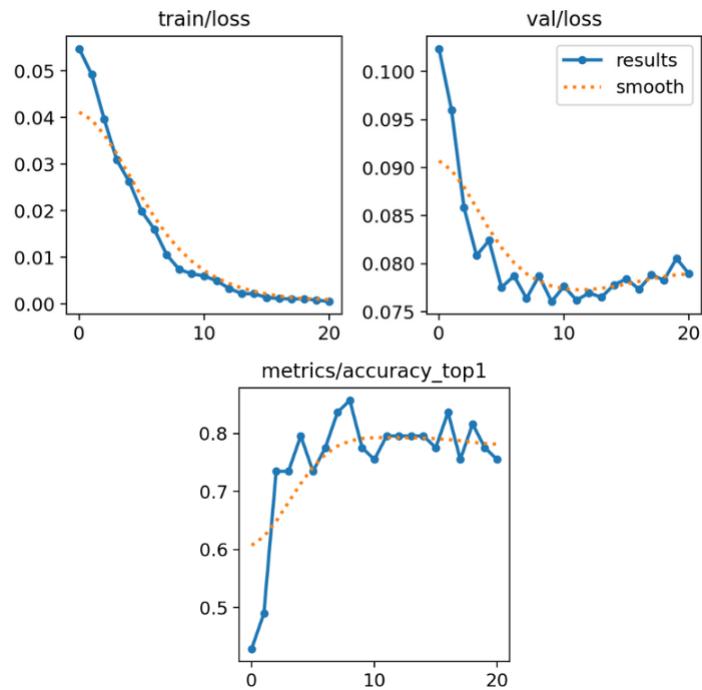


Figura 32 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE, sem a classe saudável, aplicadas sob um filtro de aumento de contraste.

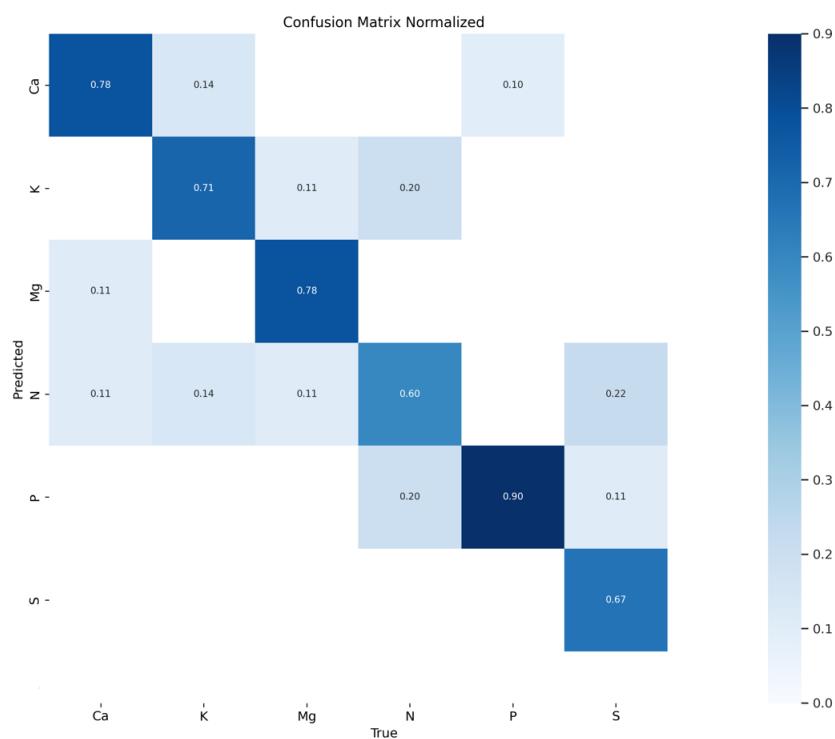


Figura 33 – Matriz de confusão dos testes realizados com as imagens selecionadas do dataset BDICAFE, sem as classe saudável, aplicadas sob um filtro de aumento de contraste.

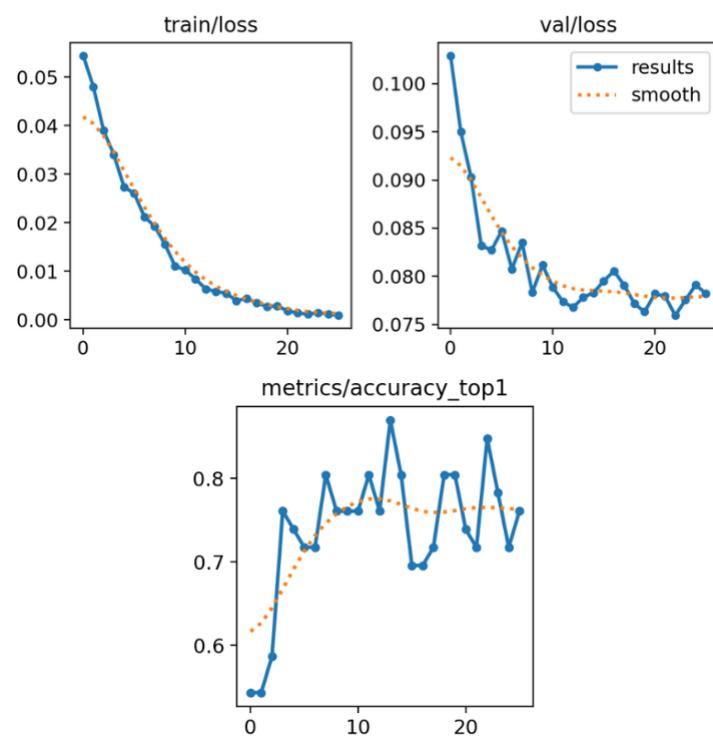


Figura 34 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE, sem a classe saudável, aplicadas sob um filtro de remoção de fundo.

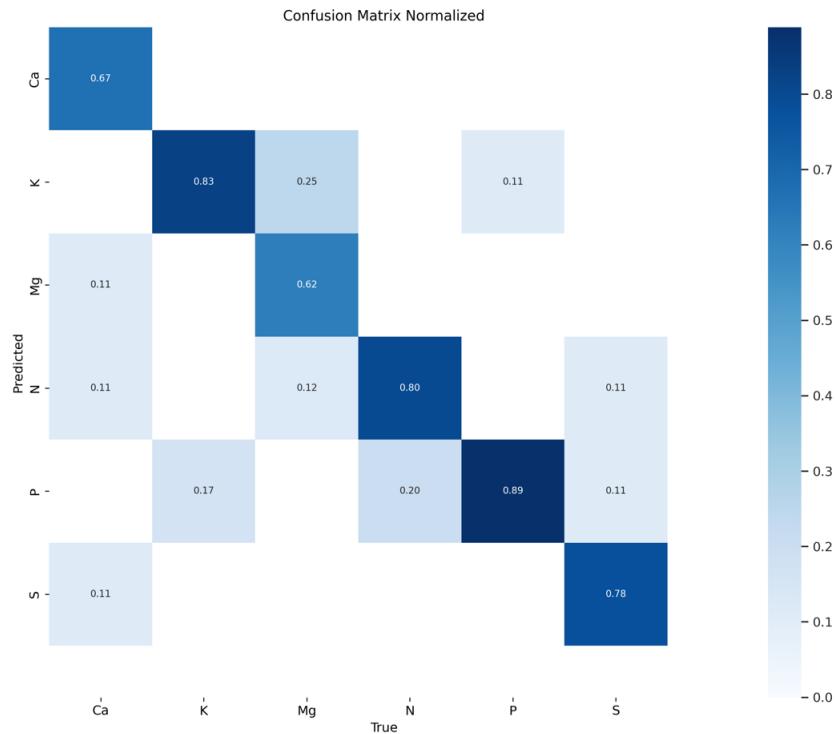


Figura 35 – Matriz de confusão dos testes realizados com as imagens selecionadas do dataset BDICAFE, sem as classe saudável, aplicadas sob um filtro de remoção de fundo.

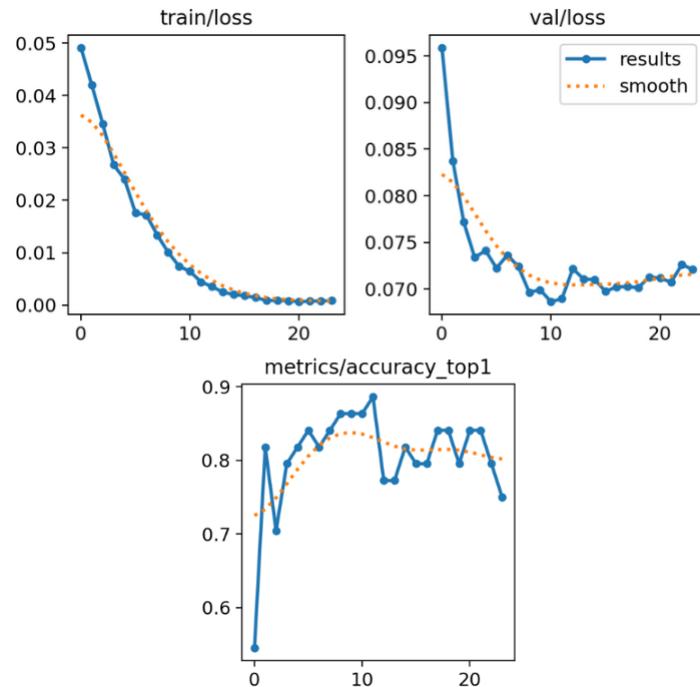


Figura 36 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE com exclusão da classe completo e nitrogênio.

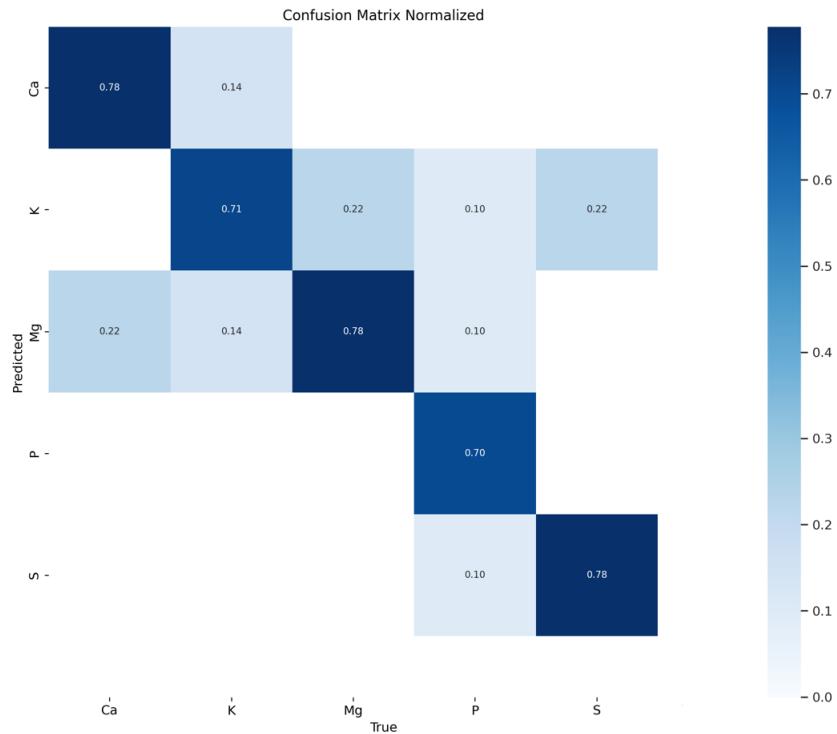


Figura 37 – Matriz de confusão dos testes realizados com as imagens selecionadas do dataset BDICAFE com exclusão da classe completo e nitrogênio.

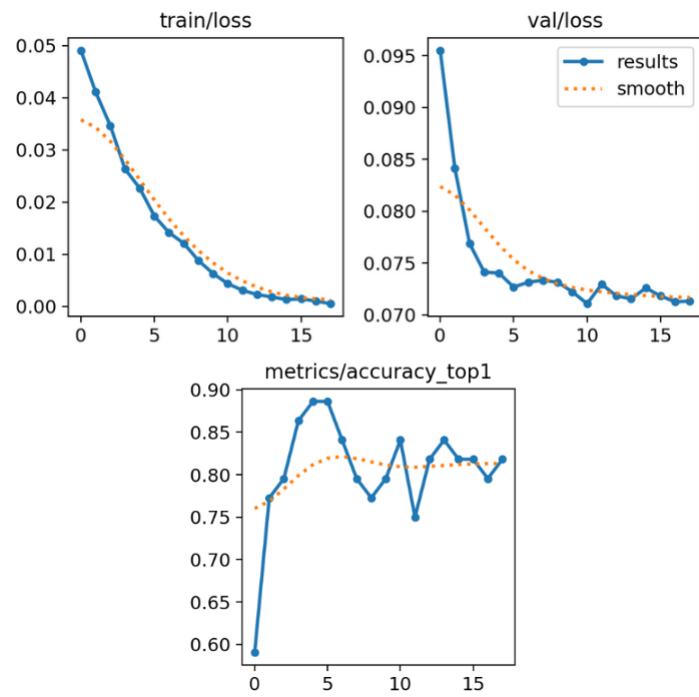


Figura 38 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE, sem as classes saudável e nitrogênio, aplicadas sob um filtro de aumento de contraste.

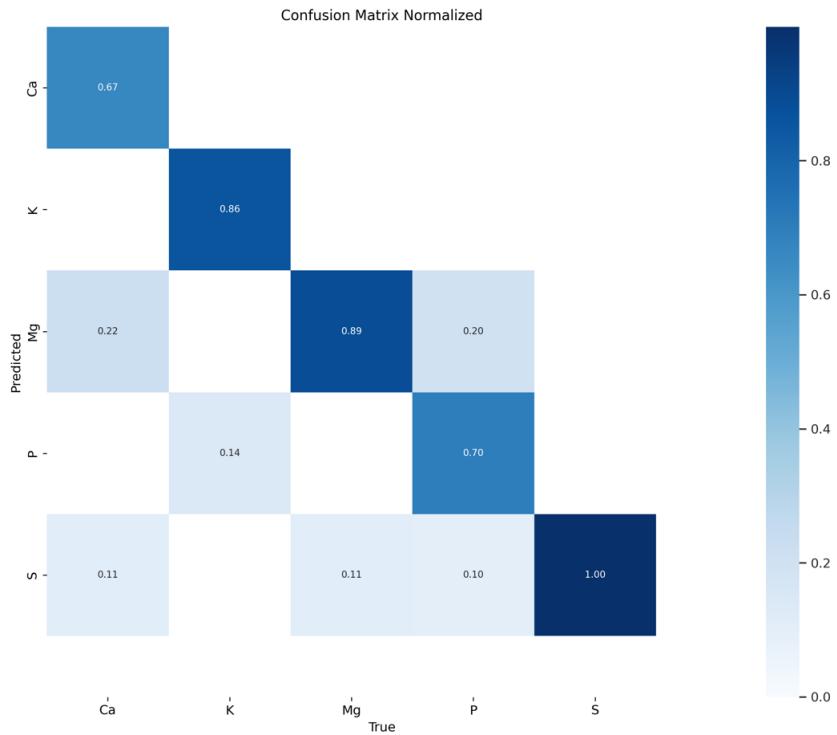


Figura 39 – Matriz de confusão dos testes realizados com as imagens selecionadas do dataset BDICAFE sem as classes saudável e nitrogênio aplicadas sob um filtro de aumento de contraste.

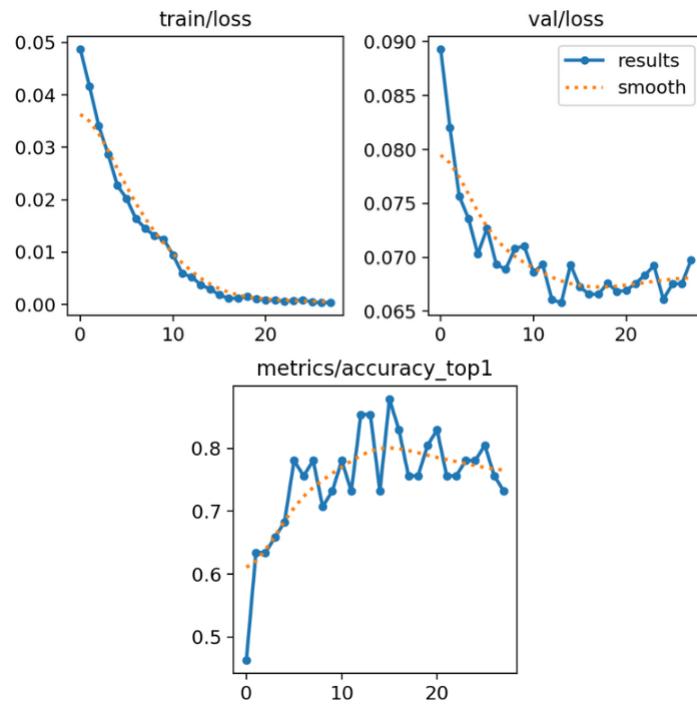


Figura 40 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE, sem as classes saudável e nitrogênio, aplicadas sob um filtro de remoção de fundo.

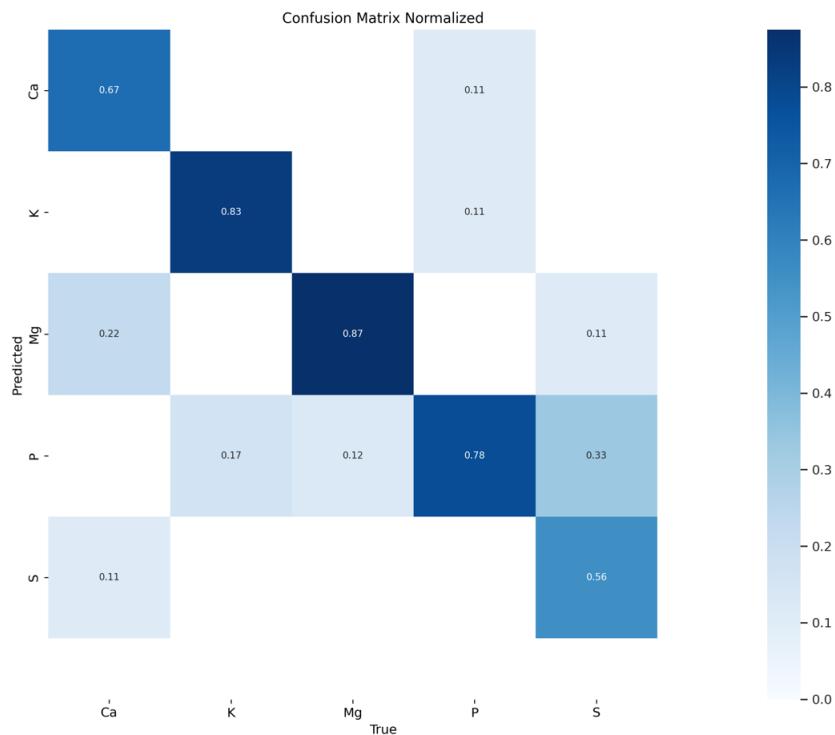


Figura 41 – Matriz de confusão dos testes realizados com as imagens selecionadas do dataset BDICAFE, sem as classe saudável e nitrogênio, aplicadas sob um filtro de remoção de fundo.

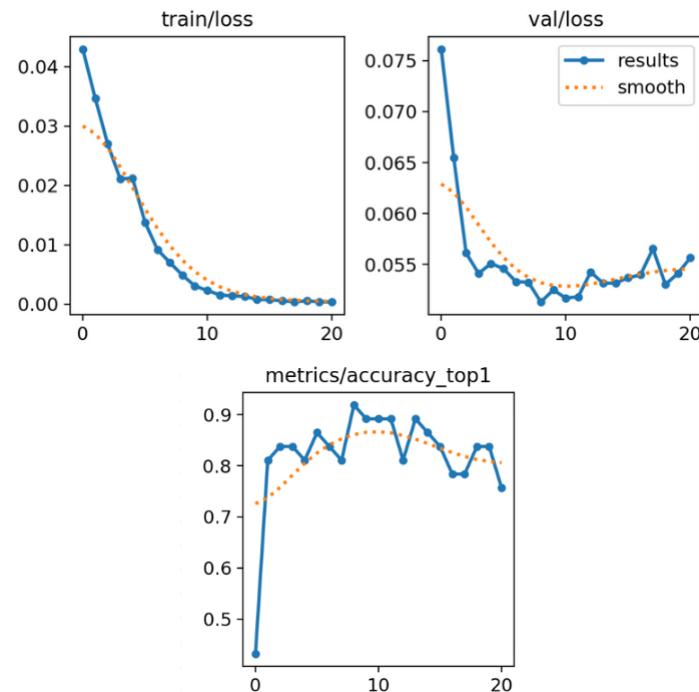
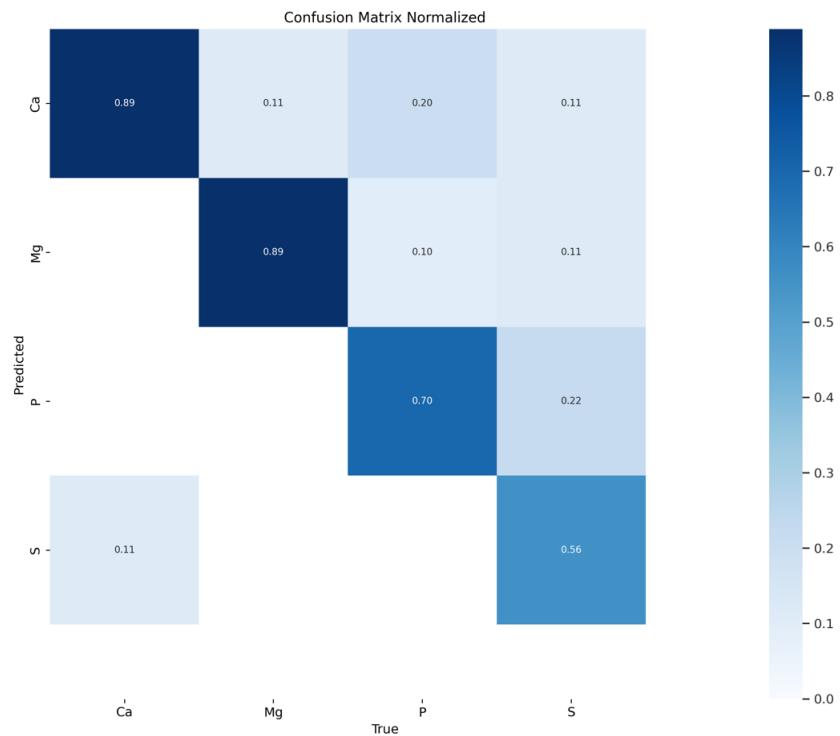


Figura 42 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE com exclusão da classe completa, nitrogênio e potássio.



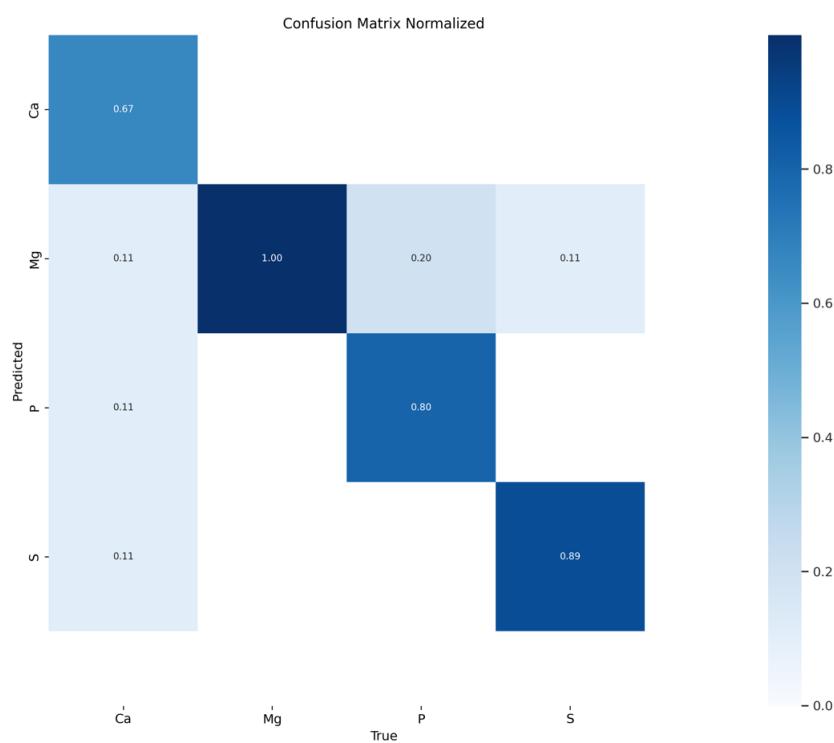


Figura 45 – Matriz de confusão dos testes realizados com as imagens selecionadas do dataset BDICAFE sem as classe saudável, nitrogênio e potássio aplicadas sob um filtro de aumento de contraste.

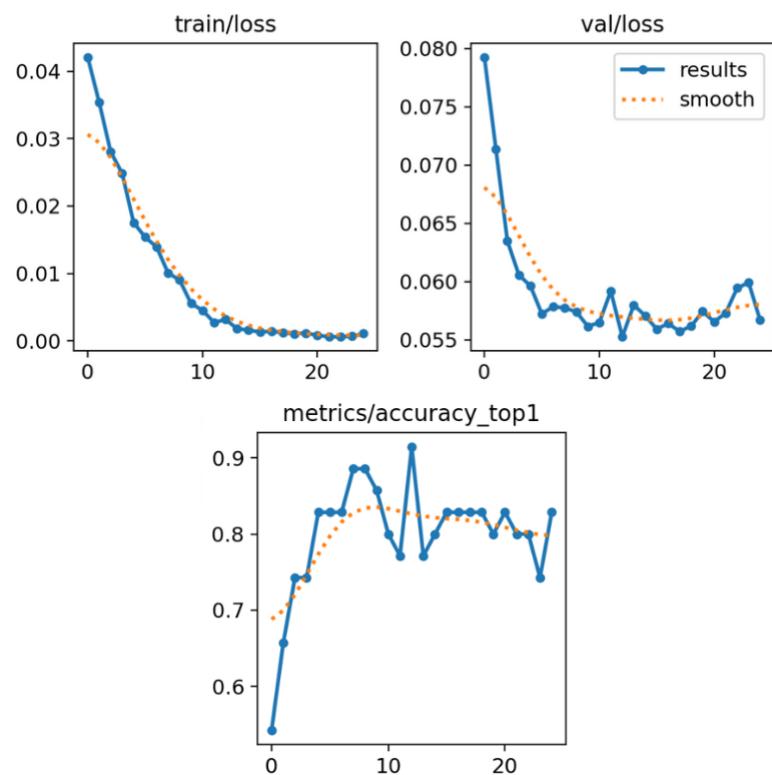


Figura 46 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE, sem a classe saudável, nitrogênio e potássio, aplicadas sob um filtro de remoção de fundo.

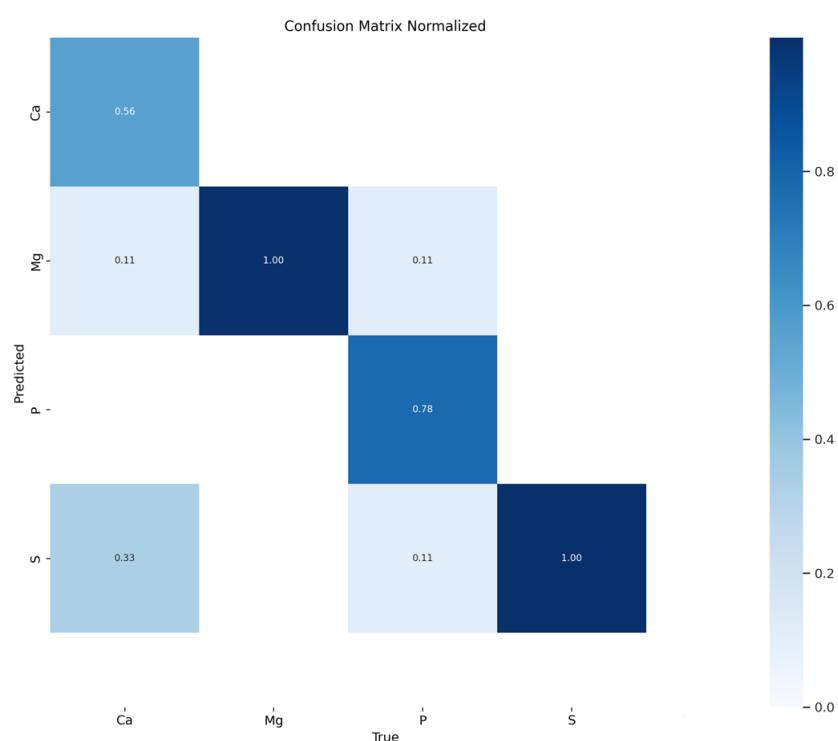


Figura 47 – Resultados do treinamento utilizando imagens selecionadas do dataset BDICAFE, sem as classes saudável, nitrogênio e potássio, aplicadas sob um filtro de remoção de fundo.