SCRIPT DE SEGMENTAÇÃO DIGITAL DE IMAGENS PARA CONTAGEM AUTOMÁTICA DE SEMENTES DE SOJA (*Glycine max*)

Maria Iza Silva de Castro[1], Maria Eduarda Lopes Fernandes[2], Caio Nepomuceno Verdeiro[2], Flaviane Mendonça Ambrozim[2], Poliany de Oliveira Barbosa[2], Patricia Alvarez Cabanez[3], Lucimara Ribeiro Venial[2],

[1]Universidade Federal do Ceará/Departamento de Fitotecnia, Avenida Mister Hull, s/n, Pici - 60455-760 - Fortaleza-CE, Brasil, mariaizasc2011@alu.ufc.br.
[2]Universidade Federal do Espírito Santo/ Programa de Pós-Graduação em Energia (PPGEN)/Laboratório LaPAQui/ Centro Universitário Norte do Espírito Santo (CEUNES), Rodovia BR-101 Norte, Km 60, Litorâneo - 29932-540 - São Mateus-ES, Brasil, mariaeduardalopesfernandes582@gmail.com, caioverdeiro@gmail.com, flaviane.ambrozim@ufes.br, poliany.barbosa@ufes.br, lucimara.venial@ufes.br.
[3]Universidade Federal Rural do Rio de Janeiro, Avenida Lourival Martins Beda, 1200, Parque Varanda do Visconde - 28178-000- Campo dos Goytacazes-RJ, Brasil, patriciacabanez@gmail.com.

Resumo
Este trabalho teve como objetivo desenvolver e realizar uma validação preliminar de um script em MATLAB, sem uso de toolboxes proprietárias, para segmentação digital e contagem automática de sementes de soja em imagens obtidas por escâner sobre fundo azul padronizado. O pipeline emprega segmentação por canais RGB, refinamento morfológico, rotulagem por flood-fill e extração de descritores geométricos e colorimétricos. O script foi avaliado em 20 imagens de cinco lotes, totalizando 2.000 sementes, e apresentou concordância integral com a contagem manual de referência, com tempo médio de 12,64 ± 0,82 s por imagem. Nas condições avaliadas, o método mostrou desempenho consistente, embora ainda deva ser testado sob maior variabilidade de aquisição e disposição das sementes.
Palavras-chave: Segmentação de imagens. Visão computacional. Sementes de soja. MATLAB. Processamento digital de imagens.

Área do Conhecimento: Ciências Exatas e da Terra / Ciência da Computação.

Introdução

A contagem e a caracterização morfométrica de sementes são etapas centrais em programas de melhoramento genético, controle de qualidade e análise de vigor em soja, sendo tradicionalmente realizadas de forma manual (Ghimire *et al.,* 2023). Esse processo, além de demorado, está sujeito à fadiga do avaliador e à variabilidade entre analistas, o que compromete a repetibilidade dos resultados em lotes numerosos, sobretudo quando são exigidas medições de grande volume de amostras em curto prazo (Mathew *et al.,* 2023).

A visão computacional e o processamento digital de imagens vêm sendo empregados como alternativas não destrutivas para automatizar a análise de sementes, permitindo ampliar o número de amostras processadas e extrair características individuais de forma padronizada (Mussadiq *et al.*, 2015; Tu *et al.*, 2023). Mussadiq *et al.,* (2015) compararam soluções de código aberto para contagem automática de sementes, evidenciando ganhos de reprodutibilidade em relação à contagem manual. Em abordagem próxima à adotada neste trabalho, Vale *et al.,* (2020) segmentaram sementes de espécies da família *Fabaceae*, incluindo soja, dispostas sobre fundo azul, obtendo separação completa entre semente e fundo sem erros de detecção. Além da contagem, algoritmos baseados em imagens RGB podem fornecer descritores de tamanho e forma, como comprimento, largura, área projetada e razão de aspecto, obtidos por Ghimire *et al.,* (2023) para sementes de soja.

Em contraste com abordagens que empregam bibliotecas consolidadas de processamento de imagens (ImageJ, OpenCV) ou modelos de aprendizado profundo, o presente trabalho adota a implementação própria das principais funções de segmentação e rotulagem em MATLAB, sem recorrer a toolboxes proprietárias.

Neste contexto, o presente trabalho tem como objetivo descrever o desenvolvimento e a arquitetura desse script de segmentação digital de imagens, voltado à contagem automática e à extração de

descritores de sementes de soja digitalizadas por escâner sobre fundo padronizado, e avaliar a consistência do seu desempenho em 20 imagens de 5 lotes, com 4 repetições cada.

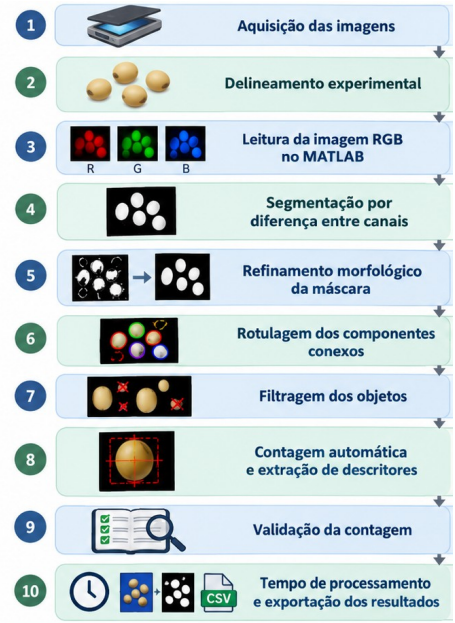
Metodologia

As sementes de soja foram dispostas manualmente, sem contato entre si, sobre fundo azul padronizado. A aquisição foi realizada por meio de um escâner HP Scanjet 2004, adaptado em posição invertida no interior de uma caixa de alumínio. As amostras foram digitalizadas a 300 dpi, utilizando o sistema de iluminação próprio do equipamento, e as imagens foram armazenadas em formato JPG. Por se tratar de aquisição por escâner, não se aplica distância câmera-amostra. Antes do processamento, cada imagem foi recortada para restringir a análise à região do fundo azul contendo as sementes.

Para avaliar a consistência do desempenho do pipeline entre diferentes lotes e repetições, foram analisados cinco lotes de sementes de soja, identificados como 02, 06, 10, 12 e 17. Cada lote foi representado por quatro repetições independentes, totalizando 20 imagens. Em cada repetição, 100 sementes foram previamente contadas manualmente antes da digitalização, totalizando 2.000 sementes. Essa contagem foi utilizada como referência para avaliação da contagem automática.

A Figura 1 apresenta o fluxo metodológico geral do processamento digital de imagens implementado no script, desde a aquisição das imagens até a validação da contagem e a geração dos arquivos de saída.

Figura 1 - Esquema metodológico do processamento digital de imagens empregado na contagem automática de sementes de soja.



Fonte: elaborado pelos autores (2026).

Etapa 1 (leitura das imagens): o script solicita a pasta principal (função uigetdir), identifica automaticamente as subpastas nela contidas, reconhece o lote e a repetição a partir do nome de cada subpasta (por exemplo, "Soja Lote 17 R4") e processa automaticamente todas as imagens presentes em cada uma delas.

Etapa 2 (segmentação por canais RGB): cada imagem é separada nos canais vermelho (R), verde (G) e azul (B), e um pixel é classificado como pertencente a uma semente quando satisfaz simultaneamente as condições $R - B > 20$ e $G - B > 0$. Esses limiares exploram o contraste entre o tegumento amarelado das sementes, com valores de R e G superiores aos de B, e o fundo azul, em que B predomina, dispensando a conversão para outros espaços de cor, como HSV, adotada por outros autores (Vale *et al.*, 2020).

Etapa 3 (máscara binária): o resultado da segmentação constitui uma máscara binária, em que o valor 1 corresponde aos pixels pertencentes às sementes e o valor 0, ao fundo. Nessa fase, a máscara ainda pode conter ruídos pontuais e pequenas imperfeições de borda.

Etapa 4 (refinamento morfológico): a máscara é refinada por abertura (erosão seguida de dilatação), que remove ruídos pontuais, e por fechamento (dilatação seguida de erosão), que preenche pequenas falhas internas, com elemento estruturante quadrado 3 × 3. As operações de erosão e dilatação foram implementadas por convolução bidimensional (função conv2), sem uso das funções prontas da Image Processing Toolbox (imerode, imdilate, imopen e imclose).

Etapa 5 (rotulagem dos componentes conexos): os objetos da máscara refinada são individualizados por um algoritmo de preenchimento por inundação (flood-fill) com pilha própria e 8-conectividade, de modo que cada conjunto de pixels conectados recebe um identificador único, sem uso das funções bwlabel, bwconncomp ou regionprops.

Etapa 6 (filtragem e extração dos descritores): objetos com área inferior a 500 pixels são eliminados como ruído residual. Para cada objeto remanescente, são calculados a área, o diâmetro equivalente, o perímetro, a circularidade, as coordenadas do centroide e os valores médios dos canais R, G e B. O diâmetro equivalente e a circularidade são obtidos, respectivamente, pelas Equações 1 e 2:

$$D_{eq} = 2\sqrt{(A/\pi)} \quad (1)$$

$$C = 4\pi A / P^2 \quad (2)$$

em que A é a área do objeto, expressa pelo número de pixels que o compõem, e P é o perímetro, estimado pelo número de pixels do objeto que possuem ao menos um vizinho de fundo na vizinhança-4.

Etapa 7 (exportação dos resultados): para cada imagem, o script gera o número de sementes identificadas, uma imagem anotada com a numeração individual das sementes, a máscara binária final e um arquivo CSV com os descritores individuais de cada semente. Além disso, são produzidos três arquivos-resumo, contendo os resultados por imagem, o resumo por repetição e a média, o desvio-padrão (DP) e o coeficiente de variação (CV) das repetições de cada lote.

Os principais parâmetros empregados no script estão apresentados na Tabela 1.

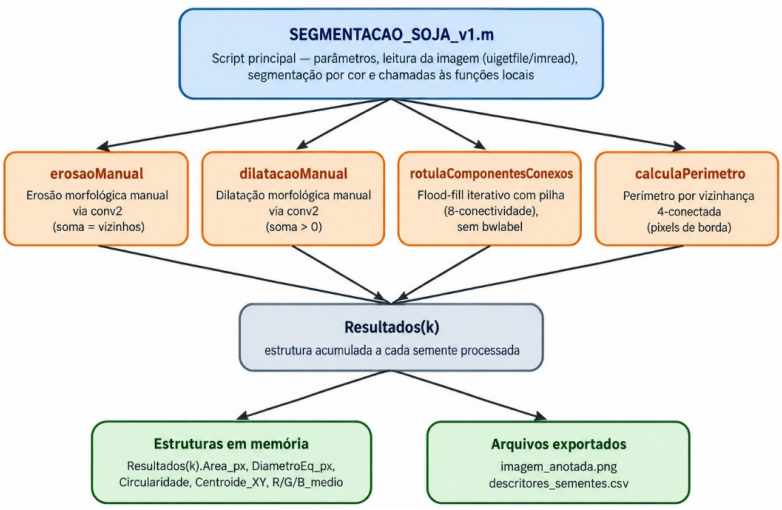
Tabela 1 - Principais parâmetros empregados no script SEGMENTACAO_SOJA_LOTES.

Parâmetro	Valor e significado
LIMIAR_RB	20 (diferença mínima R−B para o pixel ser classificado como semente)
LIMIAR_GB	0 (diferença mínima G−B para o pixel ser classificado como semente)
AREA_MIN_PX	500 px (área mínima aceita para um objeto rotulado, valores inferiores são descartados como ruído)
RAIO_ELEM_ESTRUT	1 (raio do elemento estruturante quadrado, resultando em janela 3 × 3 para a morfologia manual)

Fonte: elaborado pelos autores (2026).

A Figura 2 apresenta a arquitetura funcional do script, evidenciando a relação entre o programa principal, as funções locais de processamento e as estruturas de saída. O núcleo de processamento, derivado da versão original do script, foi mantido inalterado na versão v3, que acrescenta o processamento em lote e a medição do tempo de processamento.

Figura 2 - Arquitetura funcional do script de segmentação, incluindo programa principal, funções locais e estruturas de saída.



Fonte: elaborado pelos autores (2026).

3 A Figura 3 apresenta um trecho do núcleo de processamento implementado em MATLAB, abrangendo a separação dos canais RGB, a segmentação por diferença entre canais, o refinamento morfológico da máscara e a chamada da rotina de rotulagem dos componentes conexos.

Figura 3 - Trecho do código MATLAB correspondente às etapas de segmentação, refinamento morfológico e rotulagem dos componentes conexos.

```
I = double(I(:,:,1:3));
R = I(:,:,1);
G = I(:,:,2);
B = I(:,:,3);

diffRB = R - B;
diffGB = G - B;
Mascara = (diffRB > LIMIAR_RB) & (diffGB > LIMIAR_GB);

tam = 2*RAIO_ELEM_ESTRUT + 1;
elemEstrut = ones(tam, tam);

MascaraErodida  = erosaoManual(Mascara, elemEstrut);
MascaraAbertura = dilatacaoManual(MascaraErodida, elemEstrut);
MascaraDilatada = dilatacaoManual(MascaraAbertura, elemEstrut);
MascaraFinal    = erosaoManual(MascaraDilatada, elemEstrut);

[Rotulos, nObjetosBruto] = rotulaComponentesConexos(MascaraFinal);

Resultados = struct('ID', {}, 'Area_px', {}, 'DiametroEq_px', {}, ...
    'Circularidade', {}, 'Centroide_X', {}, 'Centroide_Y', {}, ...
    'R_medio', {}, 'G_medio', {}, 'B_medio', {});
```

Fonte: elaborado pelos autores (2026).

A validação comparou, em cada imagem, a contagem automática com a contagem manual de referência, por meio do erro absoluto, do erro relativo (em relação à contagem manual) e da acurácia de contagem. A sensibilidade (recall) e a precisão foram avaliadas em nível de objeto, com a correspondência entre sementes e objetos detectados conferida individualmente nas imagens anotadas e nas máscaras binárias, identificando-se falsos positivos (ruídos ou fragmentos contados como sementes) e falsos negativos (sementes não detectadas ou fundidas). IoU e coeficiente de Dice não foram calculados, por não haver máscara manual pixel a pixel para uso como ground truth.

O tempo de processamento foi medido individualmente para cada imagem por meio das funções tic e toc do MATLAB, abrangendo as etapas de segmentação, refinamento morfológico, rotulagem e extração dos descritores. A gravação das imagens anotadas, das máscaras e dos arquivos CSV não foi incluída no tempo medido. O processamento foi executado em notebook Lenovo IdeaPad com processador Intel Core i3.

Resultados

O script foi aplicado às 20 imagens dos 5 lotes de sementes de soja (02, 06, 10, 12 e 17), com 4 repetições cada, totalizando 2.000 sementes com contagem manual de referência. Em todas as imagens, a contagem automática coincidiu exatamente com a contagem manual de referência. Na inspeção das imagens anotadas e das máscaras binárias, não foram observadas fusões entre sementes nem fragmentações que alterassem a contagem.

A Tabela 2 apresenta a validação do script no conjunto experimental avaliado, comparando a contagem manual de referência com a contagem automática em cada lote. Em todas as 20 imagens, ambas as contagens foram de 100 sementes.

Tabela 2 - Contagem manual de referência × contagem automática de sementes de soja, por lote (4 repetições de 100 sementes por lote).

Lote	Imagens	Contagem por imagem (manual / automática)	Total manual de referência	Total automático	Erro absoluto	Erro relativo (%)
02	4	100 / 100	400	400	0	0,0
06	4	100 / 100	400	400	0	0,0
10	4	100 / 100	400	400	0	0,0
12	4	100 / 100	400	400	0	0,0
17	4	100 / 100	400	400	0	0,0
Total	**20**	**100 / 100**	**2.000**	**2.000**	**0**	**0,0**

Fonte: elaborado pelos autores (2026).

No conjunto experimental avaliado, a acurácia de contagem foi de 100% (2.000/2.000 sementes corretamente identificadas), com erro absoluto e erro relativo nulos nas 20 imagens. A inspeção das imagens anotadas e das máscaras não revelou falsos positivos nem falsos negativos, o que resultou em recall e precisão de 100% em nível de objeto. O tempo médio de processamento foi de 12,64 ± 0,82 s por imagem (mínimo de 11,01 s, máximo de 14,07 s e CV de 6,5%), totalizando 252,8 s para as 20 imagens.

Discussão

A segmentação por diferença entre canais RGB (Etapa 2) foi suficiente para separar as sementes do fundo azul homogêneo, sem conversão de espaço de cor, o que corrobora Vale *et al.,* (2020) quanto à eficácia de fundos azuis padronizados para sementes de *Fabaceae*. O refinamento morfológico (Etapa 4) garantiu a limpeza da máscara antes da rotulagem, evitando que ruídos fossem contados como sementes.

A rotulagem por flood-fill (Etapa 5) é o núcleo do pipeline de contagem, pois transforma a máscara binária em objetos individualizados. Sua implementação com pilha explícita, em vez de recursão, reduz o risco de estouro da pilha de recursão em imagens de maior resolução.

A circularidade calculada na Etapa 6 é uma aproximação, pois a contagem de pixels de borda subestima o perímetro de contornos curvos, o que explica os valores médios ligeiramente acima de 1 (1,17 a 1,20). Por isso, deve ser interpretada de forma comparativa, e não como valor absoluto (Zhu *et al.*, 2021).

Como limitação, o desempenho do script depende do contraste de cor entre fundo e sementes (Etapa 2), e alterações na iluminação ou no tom do fundo podem exigir recalibração dos limiares. Além disso, todas as imagens seguiram o mesmo protocolo de aquisição por escâner, sem variações deliberadas de iluminação, ângulo de aquisição, densidade de disposição nem sementes tangentes ou

sobrepostas, condição apontada por Zhao *et al.,* (2023) como o principal desafio para a contagem automática de sementes de soja. Em métodos baseados em componentes conexos, sementes em contato podem ser interpretadas como uma única região, e estratégias de separação, como variantes do algoritmo watershed e métodos baseados na geometria dos objetos conectados, têm sido propostas para esse problema (Sonawane; Mohanty, 2021; Liu *et al.*, 2023). Portanto, a acurácia de 100% nos 5 lotes indica consistência do pipeline, mas não permite afirmar sua generalização para outros cenários de aquisição.

Conclusão

O script desenvolvido em MATLAB apresentou potencial como ferramenta auxiliar para a contagem automatizada e a extração de descritores de sementes de soja em condições padronizadas de aquisição. Para ampliar sua aplicabilidade, ainda são necessárias avaliações sob diferentes condições de iluminação, disposição, densidade e sobreposição das sementes.

Referências

GHIMIRE, A.; KIM, S. H.; CHO, A.; JANG, N.; AHN, S.; ISLAM, M. S.; MANSOOR, S.; CHUNG, Y. S.; KIM, Y. Automatic evaluation of soybean seed traits using RGB image data and a Python algorithm. **Plants**, v. 12, n. 17, p. 3078, 2023. DOI: 10.3390/plants12173078.

LIU, H.-Q.; ZHANG, W.; WANG, F.; SUN, X.; WANG, J.; WANG, C.; WANG, X. Application of an improved watershed algorithm based on distance map reconstruction in bean image segmentation. **Heliyon**, v. 9, n. 4, art. e15097, 2023. DOI: 10.1016/j.heliyon.2023.e15097.

MATHEW, J.; DELAVARPOUR, N.; MIRANDA, C.; STENGER, J.; ZHANG, Z.; ADUTEYE, J.; FLORES, P. A novel approach to pod count estimation using a depth camera in support of soybean breeding applications. **Sensors**, v. 23, n. 14, p. 6506, 2023. DOI: 10.3390/s23146506.

MUSSADIQ, Z.; LASZLO, B.; HELYES, L.; GYURICZA, C. Evaluation and comparison of open source program solutions for automatic seed counting on digital images. **Computers and Electronics in Agriculture**, v. 117, p. 194-199, 2015.

SONAWANE, S. K.; MOHANTY, B. An improved image processing scheme for automatic detection of harvested soybean seeds. **Journal of Food Measurement and Characterization**, v. 15, 2021. DOI: 10.1007/s11694-021-01124-0.

TU, K.; WU, W.; CHENG, Y.; ZHANG, H.; XU, Y.; DONG, X.; WANG, M.; SUN, Q. AIseed: an automated image analysis software for high-throughput phenotyping and quality non-destructive testing of individual plant seeds. **Computers and Electronics in Agriculture**, v. 207, art. 107740, 2023. DOI: 10.1016/j.compag.2023.107740.

VALE, A. M. P. G.; UCCHESU, M.; DI RUBERTO, C.; LODDO, A.; SOARES, J. M.; BACCHETTA, G. A new automatic approach to seed image analysis: from acquisition to segmentation. **arXiv preprint**, arXiv:2012.06414, 2020. Preprint (não revisado por pares).

ZHAO, J.; KAGA, A.; YAMADA, T.; KOMATSU, K.; HIRATA, K.; KIKUCHI, A.; HIRAFUJI, M.; NINOMIYA, S.; GUO, W. Improved field-based soybean seed counting and localization with feature level considered. **Plant Phenomics**, v. 5, art. 0026, 2023. DOI: 10.34133/plantphenomics.0026.

ZHU, F.; PAUL, P.; HUSSAIN, W.; WALLMAN, K.; DHATT, B. K.; SANDHU, J.; IRVIN, L.; MOROTA, G.; YU, H.; WALIA, H. SeedExtractor: an open-source GUI for seed image analysis. **Frontiers in Plant Science**, v. 11, art. 581546, 2021. DOI: 10.3389/fpls.2020.581546.