

Aprimoramento de Algoritmos de Segmentação para Contagem de Bovinos em Imagens Aéreas Utilizando Processamento de Imagens

1st Gabriel Miranda

Departamento de Ciência da Computação
Universidade de Brasília
Brasília, Brasil
gabrieldaconceicaomiranda@gmail.com

2nd Máxsuel da Silva Amorim

Departamento de Ciência da Computação
Universidade de Brasília
Brasília, Brasil
maxsuelamorim@outlook.com

Abstract—This paper presents the final project for the Introduction to Image Processing course, part of the Department of Computer Science at the University of Brasília. The study focuses on image segmentation methodologies applied to cattle counting in aerial images. The project's objective is to improve upon previous results found in the literature by employing a different processing sequence and alternative algorithms [1]. The study was implemented using MATLAB.

Index Terms—k-means, image segmentation, mathematical morphology, threshold, histogram

I. RESUMO

Este artigo apresenta o trabalho final da disciplina Introdução ao Processamento de Imagens, do Departamento de Ciência da Computação da Universidade de Brasília. O estudo foca em metodologias de segmentação de imagens voltadas para a contagem de bovinos em imagens aéreas. O objetivo do projeto é aperfeiçoar os resultados obtidos em pesquisas anteriores, empregando uma sequência de processamento diferenciada e algoritmos alternativos [1]. O estudo foi implementado na linguagem MATLAB.

II. INTRODUÇÃO

O objetivo deste trabalho é aprimorar a metodologia implementada no artigo "Aplicação de algoritmos de Visão Computacional na contagem de gado por meio de processamento de imagens aéreas" [1]. Para isso, será realizada uma comparação quantitativa e qualitativa dos resultados, aplicando novas metodologias às imagens do banco de dados disponibilizado pelo artigo original.

O estudo original propôs uma abordagem em Python para a contagem de bovinos a partir de imagens aéreas, utilizando técnicas de processamento de imagens. A metodologia envolveu diversos passos, como a conversão das imagens para a escala de cinza, aplicação de filtros de suavização para redução de ruído, limiarização para definir contornos dos objetos, transformações morfológicas para refinar os resultados e, finalmente, a utilização de detectores binários para realizar a contagem. Esses procedimentos permitiram identificar e contar bovinos, mas há oportunidades para melhorias na precisão e eficiência.

Para melhor compreensão dos experimentos, é necessário que o leitor tenha um conhecimento básico sobre algumas das técnicas de processamento de imagens utilizadas. A suavização e aguçamento, por exemplo, são operações importantes para destacar ou suprimir detalhes em uma imagem. A suavização gaussiana reduz ruídos, enquanto o aguçamento, realizado com o detector de bordas Canny ou passa-alta no domínio da frequência, realça as bordas dos objetos.

Outra técnica central é a limiarização automática, especificamente o método de Otsu, que define automaticamente o melhor valor de limiar para separar os objetos de interesse do fundo da imagem. Em seguida, as transformações morfológicas, como a abertura, fechamento e preenchimento de buracos, ajudam a refinar a segmentação, corrigindo imperfeições. Além dessas técnicas, também são aplicados algoritmos mais avançados, como o Watershed para segmentação, a detecção de blobs para identificar objetos conectados e o algoritmo K-means para agrupamento de dados, visando melhorar a precisão da contagem.

A nova abordagem proposta neste trabalho visa comparar essas metodologias alternativas com os resultados obtidos no artigo anterior, buscando identificar as vantagens e limitações de cada uma.

Este relatório terá a seguinte estrutura:

- Seção III - Metodologia: Detalha os métodos e procedimentos adotados em cada etapa, incluindo a suavização da imagem, limiarização, transformações morfológicas e a aplicação do algoritmo de Watershed.
- Seção IV - Resultados: Apresenta os resultados de cada experimento com análise gráfica e discussão da eficácia das técnicas.
- Seção V - Conclusão: Resume as observações e conclusões derivadas dos experimentos e análises realizadas.

III. METODOLOGIA

A implementação do algoritmo de identificação e contagem dos animais se deu pela introdução de novas etapas de processamento ao pipeline implementado no artigo base [1], tendo como objetivo solucionar alguns problemas frequentemente

observados durante os testes do método no banco de imagens disponibilizado pelos autores do mesmo. A descrição desses problemas identificados será feita na Seção V, onde o comparativo entre as duas metodologias será exposto. A figura 1 mostra o fluxograma do novo algoritmo proposto.

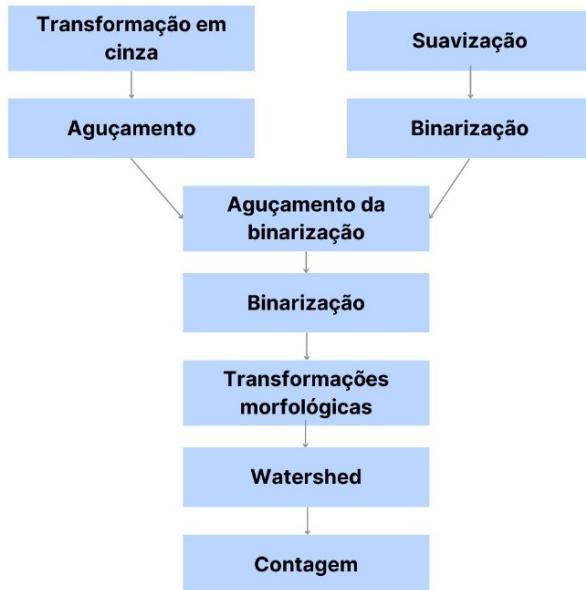


Fig. 1. Diagrama de execução.

A implementação do algoritmo é versátil, o importante são os efeitos de cada etapa de processamento e como essas interagem com o todo. Isso permite que um mesmo bloco possa ser implementado de diferentes maneiras, desde que a natureza da transformação seja a mesma.

A. Transformação em cinza

Uma das primeiras etapas no processamento é a conversão da imagem RGB em uma imagem monocromática. Além de tornar as etapas de processamento posteriores mais simples, por estar operando em apenas um canal ao invés de três, essa transformação possibilita uma aplicação futura de algoritmos de aguçamento.

B. Aguçamento

Essa etapa pode ser realizada de diferentes maneiras, sendo o principal objetivo realçar os contornos para facilitar a separação de animais próximos. No experimento, dois algoritmos distintos foram aplicados para essa tarefa.

A primeira abordagem envolveu a aplicação de uma filtragem no domínio da frequência, que tem como objetivo atenuar as componentes de baixa frequência, realçando as de alta frequência, onde geralmente estão localizados os contornos dos objetos. Ao suprimir as baixas freqüências, que correspondem às variações suaves na imagem, e destacar as bordas mais nítidas, essa técnica permite evidenciar melhor os limites dos objetos. A frequência de corte usada na filtragem

pode ser ajustada conforme a natureza das imagens, permitindo a otimização dos resultados.

A segunda abordagem foi realizada com o algoritmo Canny, um detector de bordas baseado em gradientes, que identifica de forma eficiente os contornos em imagens com ruídos leves [2]. O método Canny envolve várias etapas, como suavização da imagem com um filtro Gaussiano, cálculo do gradiente, supressão de não-máximos e a aplicação de limiar duplo. O resultado é uma detecção precisa das bordas, sendo ideal para separar objetos próximos, especialmente em condições de iluminação variada ou presença de ruídos. Comparado à filtragem no domínio da frequência, o algoritmo Canny oferece maior robustez em cenários com ruído ou variações de contraste, enquanto a filtragem por frequência é mais indicada quando as bordas estão bem representadas em altas freqüências.

C. Suavização

Essa é uma etapa paralela aos procedimentos anteriores. O objetivo deste procedimento é eliminar ruídos de alta freqüência na imagem, o que permite obter melhores resultados na etapa de processamento seguinte, a binarização.

Novamente o importante aqui é borrar a imagem de entrada e os procedimentos para tal podem variar. Nos testes utilizou-se da aplicação de um filtro Gaussiano para produzir tal efeito. Além disso, foram implementadas funções de suavização tanto para imagens RGB quanto imagens monocromáticas, permitindo que a etapa de binarização seguinte pudesse operar com dois tipos diferentes de entrada.

D. Binarização

Essa etapa é essencial para a sequência de processamento, é nela que se define o que é e o que não é um objeto. Na prática, o procedimento não é perfeito, e requer etapas posteriores para a otimização dos resultados. Existem duas binarizações na sequência proposta. A primeira pode ser implementada tanto para imagens RGB quanto para imagens monocromáticas.

Quando implementada para imagens monocromáticas, a binarização é realizada por meio de limiarização. O limiar pode ser tanto definido pelo usuário, quanto definido automaticamente pelo método de Otsu. Já para a binarização de imagens RGB, a técnica utilizada é o K-means [4]. Nesse procedimento, a imagem é dividida em classes, as classes que correspondem aos pixels dos objetos são consideradas como 1 e o restante como 0. Na maioria dos casos as classes de menores áreas são as que contêm os pixels dos animais.

Uma segunda etapa binarização também se faz presente, mas por estar mais à frente na sequência de operações essa opera apenas em imagens monocromáticas.

Alguns problemas podem surgir durante a binarização, sendo os mais frequentes união de dois objetos próximos, o que faz com que esses sejam contabilizados como apenas uma entidade, e a seleção de pixels não pertencentes ao objeto, geralmente menores e ocasionados por ruído. As etapas de processamento seguintes buscam minimizar esses importunos.

E. Aguçamento da binarização

Esta é uma etapa simples, basicamente consiste usar a primeira imagem binarizada como uma máscara para a imagem de saída da etapa de aguçamento (em que as bordas são mais escuras que o restante da imagem). Dessa forma pixels que não são objeto continuam pretos, enquanto que pixels considerados objetos podem escurecer dependendo da intensidade da borda no local. O objetivo é escurecer regiões que separam dois animais próximos, possibilitando que a próxima binarização segmente melhor as entidades.

F. Transformações Morfológicas

Operações morfológicas como abertura, fechamento, erosão, preenchimento de buracos e dilatação são usadas para eliminar pequenas regiões de pixels brancos que não são objetos de fato e tornar as delimitações das entidades mais robustas. O preenchimento de buracos nas áreas dos objetos também é outro benefício obtido.

G. Watershed

Mesmo com as etapas anteriores ainda há chance de que duas entidades permaneçam conectadas e sejam contabilizadas com apenas uma. O algoritmo Watershed é usado para otimizar separação de objetos que estão próximos uns dos outros.

Primeiro, a transformada de distância é aplicada à imagem binária para calcular a distância de cada pixel até o fundo mais próximo. Em seguida, a transformada é invertida para que os objetos sejam representados como "vales", possibilitando a segmentação pelo Watershed. Mínimos locais são suprimidos para evitar divisões excessivas. Em resumo esse método trata a imagem como uma superfície tridimensional, onde as áreas de baixa intensidade são consideradas "vales" e as de alta intensidade são consideradas "picos". O algoritmo realiza a segmentação de regiões conectadas criando linhas divisórias entre elas [3].

H. Contagem

Depois de todas as etapas de processamento, utiliza-se o Blob Analyzer do MATLAB para realizar a contagem dos animais na imagem. O Blob Analyzer permite a análise e contagem eficiente de objetos conectados em uma imagem binária, retornando informações úteis, como as áreas e as caixas delimitadoras. Além disso é possível especificar parâmetros como o tamanho mínimo para uma região ser considerada como objeto e o número máximo de objetos a serem detectados na imagem.

IV. RESULTADOS

A sequência de processamento implementada pelo artigo base apresenta alguns problemas recorrentes na identificação dos animais, dentre os quais se destacam a dificuldade em identificar animais mais escuros e o erro de contagem quando os animais estão próximos. A eficiência da nova proposta pode ser avaliada analisando a capacidade de solucionar tais problemas.

Quanto a dificuldade em identificar animais mais escuros, a binarização realizada por meio do algoritmo K-means mostrou-se uma ótima alternativa. A figura 3 mostra a divisão em classes da imagem 2. Para a binarização, costuma-se selecionar as classes de menores áreas, nesse caso 3 das 6 classes em que a imagem é dividida. Nota-se que a vaca mais escura é categorizada em uma classe diferente do fundo e portanto pode ser considerada um objeto na binarização. O resultado da binarização da figura 3 é mostrado na figura 4



Fig. 2. Imagem 2 do banco de imagens.

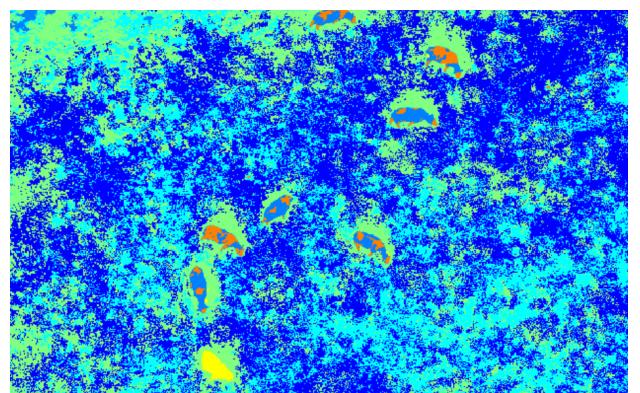


Fig. 3. Divisão da imagem 2 em classes com k-means.



Fig. 4. Binarização da imagem 2 utilizando k-means.

Quanto ao segundo problema, quando os animais estão muito próximos uns dos outros e acabam sendo considerados como uma única entidade, a etapa de aguçamento da binarização demonstrou ter uma grande importância para produzir um resultado mais preciso. A figura 5 mostra o efeito da adição dessa etapa ao pipeline original. Após a primeira binarização, as duas vacas permanecem conectadas e seriam contabilizadas juntas. Aplicando a imagem binária como máscara na imagem resultante do aguçamento, a divisão entre as duas vacas é realçada e pode ser estabelecida por meio de uma segunda binarização.

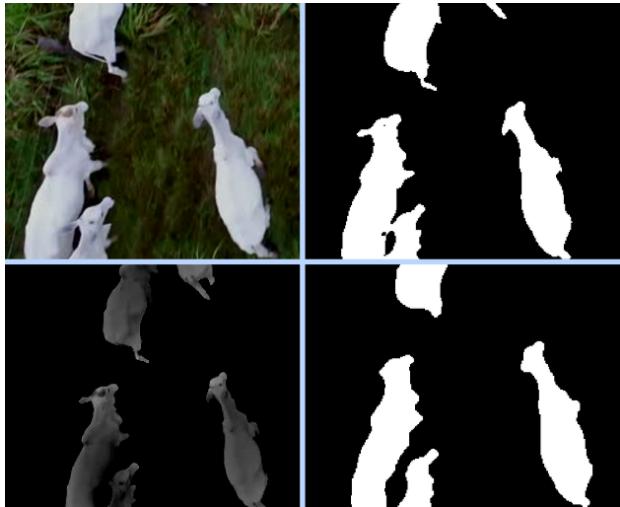


Fig. 5. Processo de aguçamento na imagem binarizada.

Acontece que nem sempre essa etapa adicional é suficiente para a distinção dos animais. A figura 6 mostra que, apesar de haver uma diminuição na magnitude do contato entre os animais, esses ainda permanecem conectados. As imagens da esquerda são resultados da primeira binarização enquanto as da direita são os resultados da segunda.

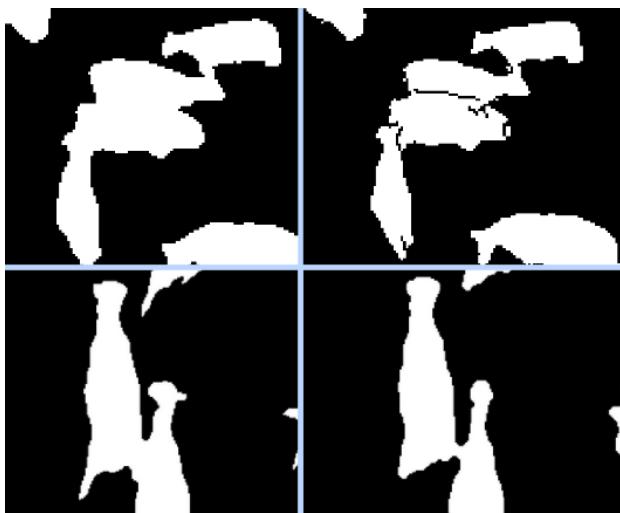


Fig. 6. Imperfeições do processo de aguçamento da imagem binarizada.

Como supracitado, a intenção é que o algoritmo Watershed possa ser usado para aperfeiçoar esse resultado. Observou-se que geralmente as alterações provocadas por esse algoritmo tornaram os resultados mais precisos. A figura 7 mostra o antes (imagens da esquerda) e depois (imagens da direita) da aplicação do algoritmo. Nota-se que as divisões entre as vacas foram completadas corretamente.

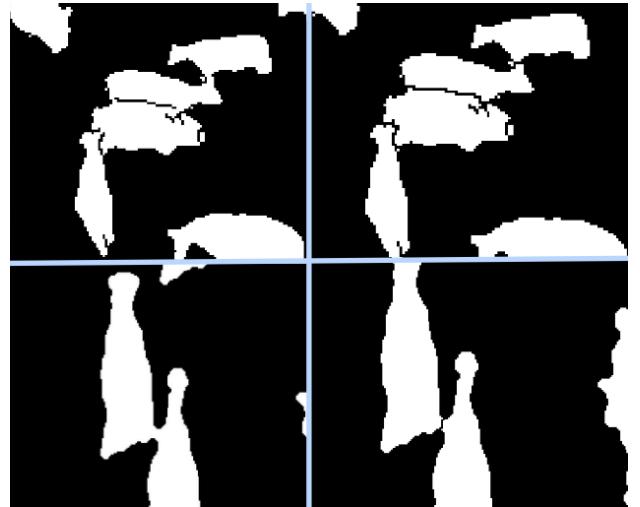


Fig. 7. Separação dos objetos utilizando o algoritmo Watershed.

Nem sempre essa sequência de etapas foi suficiente para separar os objetos. Além disso, um outro problema foi identificado. Observou-se que quando os animais eram malhados, ou apresentavam contrastes devido a iluminação, a binarização dificilmente resultava na identificação de toda a extensão dos animais, ocasionando em buracos e falhas no componente conexo gerado. Isso foi particularmente prejudicial para o algoritmo Watershed, uma vez que essas falhas eram algumas vezes "entendidas" como contornos, o que fez com que alguns objetos fossem divididos em duas entidades diferentes. A figura 8 mostra um exemplo de binarização de baixa qualidade. O resultado da aplicação do algoritmo Watershed nesse exemplo é mostrado na figura 9, nota-se que as duas vacas mostradas por completo na imagem foram divididas, sendo contabilizadas como 4 entidades.

Uma ponderação relevante a se fazer é que esse tipo de imperfeição nem sempre pode ser corrigida por meio de transformações morfológicas. A figura 8, por exemplo, é apenas um recorte da imagem original. Em outras localidades da imagem existem aglomerações de vacas que seriam unidas caso alguma operação de dilatação ou fechamento fosse utilizada para corrigir os defeitos locais apresentados. O mesmo vale para as pequenas regiões conexas brancas ocasionadas pelo ruído. Essas por sua vez podem ser ignoradas pelo algoritmo de contagem, que possibilita a escolha de uma área mínima para que uma determinada região conexa seja considerada como objeto.

Apesar de conter poucas imagens o banco de dados disponibilizado no artigo base engloba situações distintas bem capazes de ocorrerem em aplicações cotidianas. Variações de contraste



Fig. 8. Exemplo de binarização de baixa qualidade.



Fig. 9. Resultado da aplicação do algoritmo Watershed na figura 8.

entre o fundo e os animais, diferença de coloração, variações de perspectiva, a presença de filhotes, aglomerações, animais que aparecem pela metade no quadro e a presença de sombras são exemplos de possibilidades abrangidas pelas oito imagens disponibilizadas.

A figura 10 mostra o resultado da detecção feita na imagem 3 do banco de imagens. Como mostrado na parte inferior da figura 7, sem o uso do algoritmo Watershed 2 vacas da imagem 3 seriam consideradas como apenas uma, pois estariam conectadas, esse foi o caso do artigo base, em que 22 vacas de 23 foram contabilizadas. Com a adição das novas etapas de processamento é possível identificar individualmente cada vaca e obter a contagem correta.

A figura 11 mostra o resultado da detecção feita na imagem 5 do banco de imagens. Nela é possível observar sombras, aglomeração e a presença de bezerros. O algoritmo identificou todas as vacas maiores e deixou de considerar apenas 2 bezerros.

A figura 12 mostra o resultado da detecção feita na imagem 6 do banco de imagens. Nela é possível observar como a perspectiva pode variar de uma vista superior para uma vista mais frontal. Além disso, é possível notar como o tamanho dos animais pode mudar dependendo do grau de proximidade com a câmera. Isso, juntamente com a presença de filhotes,



Fig. 10. Resultado da detecção na imagem 3.



Fig. 11. Resultado da detecção na imagem 5.

dificulta a consideração das áreas no processo de segmentação.

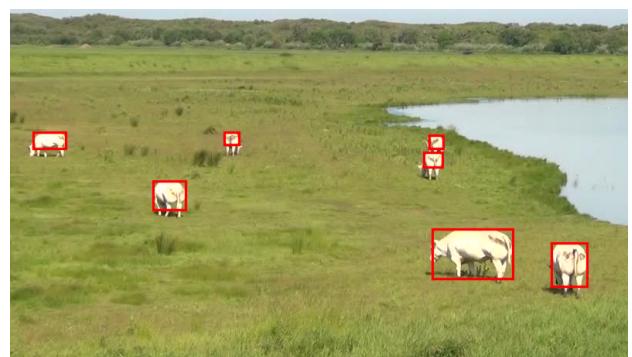


Fig. 12. Resultado da detecção na imagem 6.

A tabela I deixa evidente que a performance da detecção foi aprimorada pelas modificações propostas. No geral, o grande problema do modelo base era que ele deixava de detectar vacas de cores escuras ou vacas aglomeradas, sendo muito raro um caso de falso positivo. Já no caso do novo modelo, é mais comum a ocorrência de falsos positivos. Isso pode causar uma falsa impressão de acurácia na detecção, uma vez que uma vaca não detectada pode ser numericamente compensada por um caso de falso positivo. Apesar disso, para as imagens do

TABLE I
COMPARATIVO DE EFICIÊNCIA USANDO O BANCO DE IMAGENS
DISPONIBILIZADO NO ARTIGO BASE.

Imagen	Total de animais	Quantidade detectada no artigo base	Quantidade detectada com novo modelo
1	19	19	19
2	8	7	8
3	23	22	23
4	141	53	140
5	48	41	46
6	5	3	5
7	17	10	17
8	21	14	21

banco de dados, o único caso em que isso foi observado foi na imagem 4. Nessa imagem, 12 vacas não foram detectadas e 11 falsos positivos foram identificados, o que fez com que o número da contagem fosse 140, causando uma impressão de precisão bem maior que a realidade. Na imagem 5 não houve falsos positivos e 2 vacas não foram detectadas. O restante das imagens teve como resultado uma detecção extremamente precisa.

Tanto o método implementado no artigo base como o novo método proposto requerem a especificação manual de parâmetros, como frequências de corte, elemento estruturante, limiares e tamanho dos kernels. Isso pode ser um grande problema do ponto de vista da aplicação desses algoritmos em situações reais, já que a escolha inadequada desses parâmetros pode comprometer a qualidade dos resultados. A definição automática desses parâmetros permitiria uma adaptação mais eficiente dos algoritmos a diferentes tipos de imagens e cenários, eliminando a necessidade de intervenção humana e tornando o sistema mais robusto. Portanto, a implementação de técnicas que ajustem esses parâmetros de forma autônoma é um excelente tema para trabalhos futuros, com o potencial de aumentar significativamente a aplicabilidade prática dos métodos em ambientes dinâmicos e variados.

V. CONCLUSÃO

Com base nos resultados apresentados, é possível concluir que o novo método proposto trouxe melhorias significativas em relação ao método original descrito no artigo base, principalmente na identificação de animais escuros e na separação de animais aglomerados. As etapas adicionais, como o aguçamento da binarização e a aplicação do algoritmo Watershed, demonstraram ser eficazes para lidar com esses desafios, proporcionando uma contagem mais precisa dos animais em cenários variados. No entanto, ainda foram observadas limitações, especialmente em situações onde a binarização não capturou toda a extensão dos objetos, resultando em falhas que impactaram a segmentação.

Além disso, tanto o método original quanto o aprimorado exigem a definição manual de parâmetros, o que representa um obstáculo em sua aplicabilidade em cenários reais. O ajuste automático desses parâmetros seria um passo crucial para

aumentar a robustez e a adaptabilidade do sistema, tornando-o mais eficaz em diferentes ambientes e condições.

Portanto, para trabalhos futuros, recomenda-se a exploração de técnicas que permitam a adaptação autônoma dos parâmetros, visando melhorar a versatilidade dos algoritmos e a precisão dos resultados em um contexto prático e dinâmico.

REFERENCES

- [1] N. G. V. Ribeiro, G. B. Guedes, and T. T. Barbieri, "Aplicação de algoritmos de visão computacional na contagem de gado por meio de processamento de imagens aéreas," *Zenodo*, 2019.
- [2] L. Ding and A. Goshtasby, "On the Canny edge detector," *Pattern Recognition*, vol. 34, no. 3, pp. 721–725, 2001.
- [3] A. Körbes and R. A. Lotufo, "Análise Teórica e Prática de Segmentação de Imagens por Transformada Watershed.
- [4] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, p. 1295, 2020.