Universidade de Brasília

Faculdade de Tecnologia

Contagem e identificação individual de árvores por visão aérea

Leandro Vilela Matthias

PROJETO FINAL DE CURSO

ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Brasília

2025

Universidade de Brasília
Faculdade de Tecnologia

Contagem e identificação individual de árvores por visão aérea

Leandro Vilela Matthias

Projeto Final de Curso submetido como requisito parcial para obtenção do grau de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Dibio Leandro Borges

Brasília
2025

FICHA CATALOGRÁFICA

Matthias, Leandro Vilela.
Contagem e identificação individual de árvores por visão aérea / Leandro Vilela Matthias; orientador Dibio Leandro Borges. -- Brasília, 2025.
49 p.

Projeto Final de Curso (Engenharia de Controle e Automação) -- Universidade de Brasília, 2025.

1. Visão computacional. 2. Contagem individual de árvores. 3. Aprendizado não supervisionado. I. Borges, Dibio Leandro, orient. II. Título.

Universidade de Brasília
Faculdade de Tecnologia

Contagem e identificação individual de árvores por visão aérea

Leandro Vilela Matthias

Projeto Final de Curso submetido como requisito parcial para obtenção do grau de Engenheiro de Controle e Automação.

Trabalho aprovado. Brasília, 09 de julho de 2025:

Prof. Dr. Dibio Leandro Borges,
UnB/CIC
Orientador

Prof. Dr. Wilson Henrique Veneziano,
UnB/CIC
Examinador interno

Prof. Dr. José Maurício Santos Torres da Motta,
UnB/FT/ENM
Examinador interno

Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar engenheiros.

Agradecimentos

Agradeço ao Dr. Dibio Leandro Borges, que teve papel fundamental sendo meu orientador. Agradeço também aos meus pais e meu irmão que sempre me apoiaram na questão de me formar em engenharia mecatrônica. Quero dar um agradecimento especial ao Dr. Bruno Luiggi Macchiavello espinoza e ao Dr. Adriano Possebon Rosa, que são excelentes professores e me motivaram a gostar mais do meu curso. Por fim agradeço a todos os meus amigos que ficaram comigo nas mesinhas da FT, comeram comigo no RU e me escutaram falando de assuntos completamente aleatórios entre aulas.

“O correr da vida embrulha tudo,
a vida é assim: esquenta e esfria,
aperta e daí afrouxa,
sossega e depois desinquieta.
O que ela quer da gente é coragem.”
(Guimarães Rosa)

Resumo

Neste trabalho, é apresentado um método para a contagem e identificação individual de árvores a partir de imagens aéreas de drone, focando em contornar a necessidade de grandes volumes de dados rotulados. A abordagem se baseia em um fluxo lógico de processamento dividido em três etapas principais. A primeira etapa consiste em isolar a vegetação, onde o sistema de identificação de árvores, construído a partir de amostras indicadas pelo usuário, aprende a diferenciar e mascarar objetos não interessantes (NIOs), como solo e sombras, resultando em uma imagem contendo apenas as árvores. A segunda etapa foca na contagem individual, na qual a imagem de árvores é segmentada em regiões menores que são, em seguida, agrupadas de forma supervisionada para reconstruir cada árvore inteira, mesmo quando suas copas se tocam. Por fim, na terceira etapa, a identificação de espécies é realizada através da clusterização das árvores já contadas, agrupando-as com base em suas características visuais (como cor e textura) para estimar o número de tipos diferentes presentes na área. Os resultados da pesquisa, que comparou diversos algoritmos para cada etapa, demonstram que a abordagem é funcional, dentre os algoritmos, para clusterização foram utilizados: k-means e gaussian mixture. Dentre os algoritmos de segmentação foram testados: SLIC e ETPS.
O método também possibilita o cálculo de índices de diversidade ecológica, como Shannon e Simpson, fornecendo uma ferramenta para análise ambiental e auxílio na tomada de decisão para preservação. O trabalho conclui que a abordagem híbrida, combinando interação do usuário com processos automáticos, é uma solução viável e robusta para o problema, ainda que existam limitações como a dependência de parâmetros empíricos e a necessidade de intervenção humana em fases de treinamento.

Palavras-chave: Visão computacional; Contagem individual de árvores; Aprendizado não supervisionado.

Abstract

In this work, a method is presented for the counting and individual identification of trees from aerial drone images, focusing on circumventing the need for large volumes of labeled data. The approach is based on a logical processing workflow divided into three main stages. The first stage consists of isolating the vegetation, where the tree identification system, built from user-indicated samples, learns to differentiate and mask non-interesting objects (NIOs), such as soil and shadows, resulting in an image containing only the trees. The second stage focuses on individual counting, in which the image of trees is segmented into smaller regions that are then grouped in a supervised manner to reconstruct each entire tree, even when their canopies touch. Finally, in the third stage, the identification of species is performed through the clustering of the already counted trees, grouping them based on their visual characteristics (such as color and texture) to estimate the number of different types present in the area. The research results, which compared several algorithms for each stage, demonstrate that the approach is functional; among the algorithms, for clustering, k-means and gaussian mixture were used. Among the segmentation algorithms, SLIC and ETPS were tested. The method also enables the calculation of ecological diversity indices, such as Shannon and Simpson, providing a tool for environmental analysis and assistance in decision-making for preservation. The work concludes that the hybrid approach, combining user interaction with automatic processes, is a viable and robust solution to the problem, although limitations exist, such as the dependence on empirical parameters and the need for human intervention in training phases.

Keywords: Computer vision; Individual tree count; Unsupervised learning.

Lista de figuras

Figura 4.1 Fluxograma do sistema proposto 23
Figura 5.1 Imagem aérea de drone base 25
Figura 5.2 Máscara da imagem de testes 26
Figura 5.3 Imagem classificada com k-means 27
Figura 5.4 Imagem classificada com SLIC 27
Figura 5.5 Gráfico número de clusters versus inércia 28
Figura 5.6 Segmentos para comparação 29
Figura 5.7 Imagem com ground truth 29
Figura A.1 Segmentos com cada segmentador clusterizados com k-means 40
Figura B.1 Segmentos com cada segmentador clusterizados com DBSCAN 41
Figura C.1 Segmentos com cada segmentador clusterizados com AgglomerativeClustering 42
Figura D.1 Segmentos com cada segmentador clusterizados com OPTICS 43
Figura E.1 Segmentos com cada segmentador clusterizados com AffinityPropagation 44
Figura F.1 Segmentos com cada segmentador clusterizados com meanshift 45
Figura G.1 Segmentos com cada segmentador clusterizados com gaussianmixture . 46
Figura H.1 Segmentos com cada segmentador clusterizados com birch 47
Figura I.1 Segmentos com cada segmentador clusterizados com fuzzycmeans . . . 48
Figura J.1 Segmentos com cada segmentador clusterizados com som 49

Lista de tabelas

Tabela 4.1 Especificações do ambiente computacional. 24
Tabela 5.1 Resultado dos segmentadores 30
Tabela 5.2 Tempo dos clusters 32
Tabela 5.3 Parâmetros dos segmentadores 32

Sumário

1 Introdução Teórica 13
1.1 Aprendizado não supervisionado - clusterização 13
1.1.1 K-means 13
1.1.2 DBSCAN 13
1.1.3 Agglomerative Clustering 14
1.1.4 OPTICS 14
1.1.5 Affinity Propagation 14
1.1.6 Mean Shift 15
1.1.7 Birch 15
1.1.8 Fuzzy C-Means 15
1.1.9 SOM 15
1.2 Aprendizado não supervisionado - segmentação 15
1.2.1 SLIC 16
1.2.2 ETPS 16
1.2.3 SEEDS 16
1.2.4 ERS 16
1.2.5 CRS 17
1.3 Aprendizado supervisionado 17
1.3.1 k-NN 17
1.3.2 Matriz de métrica para pesos 17
1.4 Diversidade 18

2 Objetivos 19
2.1 Objetivo Geral 19
2.2 Objetivos Específicos 19

3 Premissas do Trabalho 20

4 Metodologia 21
4.1 Identificação de árvores 21
4.2 Contagem individual de árvores 22
4.3 Identificação espécies 22
4.4 Fluxo lógico do sistema 22
4.5 Extração de Indicadores Ecológicos 24
4.6 Recursos Computacionais 24

5 Resultados 25
5.1 Detecção de NIO's 25

5.2 Detecção de árvores . . . 26
5.3 Detecção de espécies . . . 26
5.4 Comparação de resultados - segmentação . . . 28
5.5 Comparação de resultados - clusterização . . . 30
5.6 Índices de Diversidade . . . 31
5.7 Parâmetros utilizados . . . 32

6 Discussão . . . 33
6.1 Desempenho na detecção de NIOs . . . 33
6.2 Desempenho na segmentação . . . 33
6.3 Desempenho na clusterização . . . 33
6.4 Implicações dos resultados . . . 34
6.5 Parâmetros . . . 35
6.6 Limitações . . . 35

7 Conclusão . . . 36

8 Propostas para Estudo Futuro . . . 37
8.1 Melhorias possíveis para o mascaramento . . . 37
8.2 Melhorias possíveis para segmentação . . . 37
8.3 Melhorias possíveis para clusterização . . . 37

Referências . . . 38

Apêndice A k-means . . . 40

Apêndice B DBSCAN . . . 41

Apêndice C Agglomerative Clustering . . . 42

Apêndice D OPTICS . . . 43

Apêndice E Affinity Propagation . . . 44

Apêndice F Mean-Shift . . . 45

Apêndice G Gaussian Mixture . . . 46

Apêndice H Birch . . . 47

Apêndice I Fuzzy C-Means . . . 48

Apêndice J SOM . . . 49

1 Introdução Teórica

A contagem individual de árvores (ITS) é um problema antigo sem uma solução ótima, pois, além de haver uma variedade exorbitante de possibilidades, entre elas posição, forma, sobreposição, coloração, inclinação e espécies, conseguir dados com *outputs validados* é um grande problema também. A ausência de dados identificados torna o aprendizado de máquina não supervisionado uma abordagem promissora para resolver o problema.

1.1 Aprendizado não supervisionado - clusterização

A obtenção de imagens rotuladas em grande quantidade e com qualidade é um desafio. Nesse cenário, o aprendizado não supervisionado surge como uma forte solução. Existem diversos tipos de treinamentos não supervisionados para inteligências artificiais, sendo os mais comuns: agrupamento (clustering), associação e redução de dimensionalidade.

Neste contexto, foco será principalmente no agrupamento (clustering). O objetivo é identificar pixels ou superpixels que pertencem às mesmas árvores, e para isso, podem ser utilizadas suas características para agrupá-las.

1.1.1 K-means

Um dos algoritmos de agrupamento mais antigos e conhecidos é o k-means. Este algoritmo produz ótimos resultados para conjuntos de dados em que objetos semelhantes possuem características semelhantes. Para utilizá-lo, é necessário fornecer como entrada a quantidade de clusters desejados no conjunto de dados analisado. Com essa informação, o algoritmo classifica cada dado em um cluster, utilizando a distância de cada ponto ao centro do cluster (centroide). O k-means calcula a distância de cada ponto aos centroides, ajusta os centros e recalcula as distâncias iterativamente. Após várias iterações, o processo se estabiliza, e a menor distância a um centroide define o cluster ao qual o objeto pertence (MacQueen, 1967).

1.1.2 DBSCAN

O DBSCAN (Density-Based Spatial Clustering of Applications with Noise) é um algoritmo de agrupamento baseado em densidade que, ao contrário do k-means, não exige a definição prévia do número de clusters. Ele agrupa pontos próximos entre si, formando regiões densas, e identifica pontos isolados como ruído. O algoritmo utiliza dois parâmetros principais: ε, que define a distância máxima entre pontos vizinhos, e minPts, que é o número mínimo de pontos necessários para formar um cluster. O processo inicia com a seleção

aleatória de um ponto. Se ele possui vizinhos suficientes dentro da distância ε, é marcado como parte de um novo cluster, que se expande adicionando novos pontos que atendam ao critério de densidade. Caso um ponto não tenha vizinhos suficientes, ele é classificado como ruído. Esse processo se repete até que todos os pontos sejam classificados. O DBSCAN é altamente eficiente para detectar clusters de formatos irregulares e lidar com dados ruidosos, sendo amplamente utilizado em análise de padrões e detecção de anomalias (Ester *et al.*, 1996).

1.1.3 Agglomerative Clustering

O Agglomerative Clustering é um método hierárquico de agrupamento que segue uma abordagem bottom-up. Inicialmente, cada ponto é considerado um cluster individual. Em cada iteração, os dois clusters mais próximos são fundidos com base em uma métrica de distância (como a média das distâncias, mínima distância ou distância entre centroides). O processo continua até que todos os pontos estejam em um único cluster ou se atinja um número desejado de clusters. O resultado é frequentemente representado como um dendrograma, permitindo uma visualização hierárquica da formação dos grupos (Müllner, 2011).

1.1.4 OPTICS

O OPTICS (Ordering Points To Identify the Clustering Structure) é um algoritmo que, assim como o DBSCAN, se baseia na densidade. Ele não forma explicitamente clusters, mas cria uma ordenação dos pontos que representa a estrutura de densidade dos dados. Isso permite identificar clusters com diferentes densidades sem a necessidade de fixar um valor de *eps*. O OPTICS é útil para explorar dados complexos e visualizar hierarquias de densidade por meio de um gráfico de alcance (reachability plot) (Ankerst *et al.*, 1999).

1.1.5 Affinity Propagation

O Affinity Propagation é um algoritmo de agrupamento que não exige a definição prévia do número de clusters. Ele funciona trocando mensagens entre pares de pontos até que surjam representantes ideais, chamados de *exemplars*. As mensagens representam a aptidão de um ponto para ser exemplar e a adequação de um ponto pertencer a outro. A afinidade entre pontos é baseada em similaridade, geralmente calculada pela distância negativa. Ao final das iterações, os clusters são formados em torno dos exemplars escolhidos (Friedman; Hastie; Tibshirani, 2008).

1.1.6 Mean Shift

O Mean Shift é um algoritmo baseado em densidade que identifica os modos (picos de densidade) no espaço dos dados. Ele opera movendo iterativamente uma janela sobre os dados em direção à região de maior densidade, recalculando a média dos pontos dentro da janela. Ao final, os pontos convergem para modos locais, que se tornam os centros dos clusters. O número de clusters não precisa ser definido previamente, mas a largura da janela (ou banda) afeta diretamente a sensibilidade do algoritmo (Cheng, 1995).

1.1.7 Birch

O BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) é um algoritmo projetado para lidar com grandes volumes de dados. Ele constrói uma estrutura hierárquica chamada CF Tree (Clustering Feature Tree), que resume os dados em subclusters compactos. Depois, aplica um método de agrupamento (como k-means) nos subclusters para formar os clusters finais. O BIRCH é eficiente em termos de tempo e memória e funciona bem com dados que cabem parcialmente na memória (Zhang; Ramakrishnan; Livny, 1996).

1.1.8 Fuzzy C-Means

O Fuzzy C-Means é uma extensão do k-means baseada em lógica fuzzy. Em vez de atribuir um ponto exclusivamente a um cluster, ele calcula um grau de pertencimento de cada ponto a todos os clusters, representado por uma matriz de probabilidades. O algoritmo ajusta iterativamente os centróides e os graus de associação com base na distância dos pontos. Esse método é útil quando os dados têm fronteiras de cluster não nítidas e onde é desejável que os pontos pertençam parcialmente a múltiplos clusters (Davé; Krishnapuram, 1997).

1.1.9 SOM

O SOM (Self-Organizing Map) é uma técnica de agrupamento e visualização baseada em redes neurais não supervisionadas. Ele projeta dados de alta dimensão em uma grade de duas dimensões preservando a topologia dos dados. Durante o treinamento, cada vetor de entrada é comparado com os nós da grade, e o mais semelhante (neurônio vencedor) e seus vizinhos são ajustados para se aproximar do vetor. Isso resulta em uma representação espacial dos dados em que proximidade na grade indica similaridade entre os dados (Kohonen, 1995).

1.2 Aprendizado não supervisionado - segmentação

Um algoritmo de segmentação é um algoritmo de clustering adaptado para segmentar imagens em regiões que pertencem a um mesmo tipo. Sendo assim, a grande diferença de

um segmentador e um cluster é que o segmentador leva a posição da imagem como um dado, além disso permite que dados próximos tenham influência na classificação do pixel classificado.

1.2.1 SLIC

O SLIC (Simple Linear Iterative Clustering) é um algoritmo de segmentação que divide uma imagem em superpixels — regiões de pixels com características semelhantes — utilizando uma versão modificada do k-means. Ele opera em um espaço 5D que combina informações de cor (CIELAB) e posição espacial (XY), agrupando pixels com base em uma distância que pondera cor e proximidade espacial. O algoritmo inicia com centros de clusters distribuídos regularmente, associa pixels aos centros mais próximos e atualiza iterativamente os centros até a convergência. Após a segmentação, uma etapa de pós-processamento assegura a conectividade dos superpixels (Achanta *et al.*, 2012).

1.2.2 ETPS

O ETPS (Extended Topology Preserving Segmentation) é um algoritmo de segmentação que visa preservar a topologia das estruturas presentes na imagem. Ele utiliza uma abordagem baseada em grafos para garantir que as relações espaciais entre os pixels sejam mantidas durante a segmentação. O ETPS é particularmente eficaz na segmentação de imagens com estruturas complexas, como imagens de satélite ou médicas, onde a preservação da topologia é crucial para a análise subsequente (Yao *et al.*, 2015).

1.2.3 SEEDS

O SEEDS (Superpixels Extracted via Energy-Driven Sampling) é um algoritmo de segmentação que utiliza uma abordagem de otimização baseada em energia para extrair superpixels. Ele começa com uma segmentação inicial da imagem e, em seguida, refina iterativamente os limites dos superpixels para minimizar uma função de energia que considera a homogeneidade de cor e a regularidade espacial. O SEEDS é conhecido por sua eficiência computacional, sendo capaz de operar em tempo real, o que o torna adequado para aplicações que exigem processamento rápido (Bergh *et al.*, 2012).

1.2.4 ERS

O ERS (Entropy Rate Superpixel Segmentation) é um algoritmo que modela a segmentação como um problema de otimização baseado na taxa de entropia de uma caminhada aleatória em um grafo. Ele busca uma segmentação que maximize a entropia, promovendo superpixels compactos e com fronteiras bem definidas. O ERS é eficaz na preservação das

bordas dos objetos e na geração de superpixels de tamanhos semelhantes, sendo amplamente utilizado em tarefas que requerem segmentações precisas (Liu *et al.*, 2011).

1.2.5 CRS

O CRS (Contour Relaxed Superpixels) é um algoritmo de segmentação que relaxa as restrições de contorno para permitir a formação de superpixels mais adaptativos. Ele utiliza uma abordagem que equilibra a aderência às bordas dos objetos com a regularidade dos superpixels, permitindo uma segmentação mais flexível que pode se adaptar melhor a diferentes estruturas na imagem. O CRS é particularmente útil em cenários onde as fronteiras dos objetos não são bem definidas ou são complexas (Mester; Conrad; Guevara, 2011).

1.3 Aprendizado supervisionado

Para tarefas mais específicas, mesmo com poucos dados é possível trabalhar com uma inteligência artificial com aprendizado supervisionado, pois para distinguir um objeto muito diferente do outro muitas vezes basta uma pequena quantidade de amostrar para traçar uma diferença que se extenderá para diversos exemplos.

1.3.1 k-NN

O algoritmo k-NN (k-Nearest Neighbors), ou k-Vizinhos Mais Próximos, é um método simples e intuitivo usado para classificação e regressão em aprendizado de máquina. Ele se baseia na ideia de que objetos semelhantes tendem a estar próximos no espaço de características. O funcionamento do k-NN envolve a escolha de um valor para k, que define o número de vizinhos mais próximos a serem considerados. Para um novo ponto de dados, o algoritmo calcula a distância entre ele e todos os pontos do conjunto de treinamento, selecionando os k vizinhos mais próximos. Em tarefas de classificação, o novo ponto é atribuído à classe mais frequente entre os vizinhos; em regressão, é calculada a média ou mediana dos valores dos vizinhos. O k-NN é simples, adaptável e não faz suposições sobre a distribuição dos dados, mas pode ser lento em grandes conjuntos de dados e sensível à dimensionalidade e à escala das características. Ele é amplamente aplicado em reconhecimento de padrões e sistemas de recomendação (Fix; Hodges, 1951).

1.3.2 Matriz de métrica para pesos

O método busca otimizar a segmentação de dados aprendendo uma métrica específica. Os dados de treinamento são representados pela matriz X (tamanho $n \times d$) e a matriz de atribuições Y (tamanho $n \times K$), onde K é o número de clusters. O centro de cada cluster é dado por Z, calculado como:

$$Z_j = \frac{\sum_i Y_{ij} x_i}{\sum_i Y_{ij}}. \tag{1.1}$$

O objetivo é aprender uma métrica M que minimize o erro quadrático médio (MSE) entre os dados e os centros dos clusters, conforme a equação:

$$\min_M ||X - YZ||_M^2. \tag{1.2}$$

A solução para essa otimização é M_1, que é ajustada para obter uma matriz simétrica positiva semidefinida. A métrica M desejada é obtida por uma matriz de transformação P, conforme:

$$M = PM_1P^T. \tag{1.3}$$

O parâmetro ε controla o ajuste entre as características, como forma e cor, na segmentação (Wang *et al.*, 2023).

1.4 Diversidade

Para identificar a diversidade de uma vegetação e distinguir qual área é mais adequada para preservação, pode-se utilizar métricas objetivas de decisão. Exemplos comuns de indicadores de diversidade são os índices de Simpson e Shannon. Ambos produzem valores calculados com base no número total de indivíduos (como árvores) e na quantidade de espécies identificadas.

O índice de Simpson (Simpson, 1949) pode ser calculado por:

$$D = \sum_{i=1}^{s} \left[\frac{n_i(n_i - 1)}{N(N - 1)} \right], \tag{1.4}$$

onde n_i é o número de indivíduos da espécie i, N é o número total de indivíduos, e s é o número total de espécies. O valor de D varia entre 0 e 1, sendo que valores mais próximos de 0 indicam maior diversidade. A forma complementar $1 - D$ também é comumente usada, pois valores mais próximos de 1 indicam maior diversidade.

O índice de Shannon (Shannon, 1948) é dado por:

$$H' = -\sum_{i=1}^{s} (p_i)(\log_2 p_i), \tag{1.5}$$

em que p_i representa a proporção de indivíduos da espécie i em relação ao total. Diferente do índice de Simpson, o valor de H' não é normalizado, mas quanto maior for seu valor, maior é a diversidade da comunidade.

2 Objetivos

2.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver uma solução computacional capaz de, a partir de uma única imagem aérea obtida por drone, realizar a contagem de árvores individuais e estimar a quantidade de espécies diferentes presentes na área fotografada.

2.2 Objetivos Específicos

Para atingir este propósito, os seguintes objetivos específicos foram traçados, focados na criação de um software com funcionalidades claras:

1. **Desenvolver um software para identificação de árvores individuais:** O programa deve ser capaz de receber uma imagem aérea como entrada e, através de um pipeline de processamento, devolver os segmentos que representam cada árvore. Este processo inclui a separação da vegetação de outros objetos na imagem e o agrupamento de superpixels para formar as árvores completas.
2. **Implementar uma funcionalidade para classificação de espécies:** A partir das árvores já identificadas, o software deve aplicar algoritmos de clusterização para agrupar indivíduos com características visuais semelhantes. O resultado deve ser uma nova segmentação da imagem, onde cada grupo de cor representa uma potencial espécie de árvore.
3. **Integrar o cálculo de um índice ecológico:** O software deve utilizar os dados gerados nas etapas anteriores (número total de árvores e número de espécies) para calcular e apresentar um índice de diversidade ecológica, como o de Shannon ou Simpson. Isso transforma a ferramenta em um instrumento prático para análise ambiental.

3 Premissas do Trabalho

Para prosseguir com o trabalho, é fundamental que algumas hipóteses sejam postas como base. Para tal, será tomado como verdade que:

1. O chão ou qualquer outra estrutura que não seja árvore tem uma característica de cor bem definida pixel a pixel.
2. Árvores que que possam ter várias copas separadas podem ser contadas como árvores diferentes.
3. A distância do chão da imagem é aproximadamente constante.
4. O período dos(as) dados/imagens de treinamento é próximo do que será utilizado na análise futura.

Para entender melhor cada colocação serpa explicado o fundamento de cada um a seguir:

1. É preciso separar árvores do que não é árvore para ter uma identificação melhor sobre cada árvore individualmente.
2. Computacionalmente é muito difícil decidir se uma árvore é realmente a mesma em um lado e no outro, o que por sua vez poderia exigir uma rede mais complicada com muito mais dados para treinamento como uma cnn (convolutional neural network).
3. Para conseguir uma boa base de parâmetros internos do sistema, é melhor que o tamanho mínimo de uma árvore na imagem seja aproximadamente identificável com relação à quantidade de pixels que ocupa, diminuindo a chance de contar um aglomerado de árvores pequenas como apenas uma.
4. A cor das árvores e suas assinaturas espectrais podem mudar conforme a estação do ano, isto pode vir a ser um grande problema. Identificar árvores no período de seca pode envolver identificar árvores sem copas, inviabilizando os métodos pensados para este trabalho.

4 Metodologia

Como o objetivo deste trabalho é conseguir separar todas as árvores individualmente e em seguida verificar quais são da mesma espécie, o projeto é dividido em três etapas principais:

- Isolar árvores e não árvores;
- Separar árvores diferentes e
- Identificar árvores com características parecidas.

Então é obtida a seguinte ordem de dados:

1. A imagem bruta deve ser processada inicialmente para identificar pixel a pixel o que pertence ou não a uma árvore.
2. Pegar os dados que representam as árvores e segmentar de maneira que cada árvore possa ser identificada individualmente. Os dados que representam cada árvore deve ser um conjunto de pixels com labels únicos, de tal forma que o total de labels únicos deve ser o total de árvores na imagem.
3. A partir das árvores individuais, pode-se então tirar os dados característicos de cada uma das árvores, esses dados devem ser comparados com os dados das outras árvores da mesma imagem, identificando quais são semelhantes entre si, de tal forma que a quantidade de tipos encontrados deve ser decidido de acordo com os dados.

4.1 Identificação de árvores

Para conseguir identificar árvores e tirar o fundo da imagem, ou objetos não interessantes (NIO), será utilizado o k-NN, pois ele consegue identificar variações grandes de cor e pode ser treinado com uma quantidade pequena de dados, mas será utilizada a versão KD-tree (Bentley, 1975) para melhorar o desempenho. Para o treinamento do k-NN será utilizado o método de (Wang *et al.*, 2023), onde o usuário deve identificar em uma imagem pontos de interesse (árvore) e pontos de não interesse (chão, água, sombras, etc.), a partir dessas amostras uma k-NN será treinada para identificar pixels que devem ser avaliados posteriormente ou ignorados.

Para melhorar o desempenho e velocidade de identificação posterior, uma LUT (look up table) pode ser utilizada para mudar o tipo de consulta, no lugar de calcular as distâncias internamente em uma k-NN, olha-se uma tabela com todas as possibilidades de cores (256^3). Esta LUT é preenchida pela k-NN treinada pelo usuário.

4.2 Contagem individual de árvores

A individualização de árvores em aglomerados densos requer critérios de separação mais refinados, uma vez que o uso de metadados se mostra insuficiente. Portanto, esta etapa emprega algoritmos de segmentação de imagem para isolar cada espécime. O processo é aplicado exclusivamente sobre as áreas classificadas como ‘árvore‘ na etapa anterior, uma metodologia similar à de (Wang *et al.*, 2023). Serão testados diversos segmentadores, priorizando aqueles com melhor avaliação no artigo "Superpixels: An evaluation of the state-of-the-art"(Stutz; Hermans; Leibe, 2018). Para a execução dos testes, foi utilizada a plataforma Docker, disponível no repositório do autor: github.com/davidstutz/superpixel-benchmark.

A partir da separação das copas, como há uma grande variedade de tamanho de árvores, podendo então agrupar os segmentos que na verdade pertencem a mesma árvore, como também é visto em (Wang *et al.*, 2023). Treinando uma matriz de métrica, deve-se olhar segmento a segmento e, dado um threshold, identificar quais segmentos devem ser juntados.

4.3 Identificação espécies

A separação de espécies pode ser abordada de diversas maneiras, neste caso será feito um método de clusterização. Árvores de mesma espécie tender a ser semelhantes entre si, então serão usados diversos algoritmos de clustering, escolhidos de acordo seguindo as implementações disponíveis em python e sua aparição no artigo "Survey of Clustering Algorithms"(Xu; II, 2005). Para identificar cada árvore serão usados dados como canais de cor A e B (do LAB, ignorando L) e os desvios padões dentro de cada segmento.

4.4 Fluxo lógico do sistema

O fluxo completo da técnica proposta pode ser visualizado na Figura 4.1.

O sistema completo é composto por etapas automáticas, indicadas em verde, que podem ser executadas sem intervenção humana; etapas interativas de treinamento com o usuário, indicadas em amarelo; pontos de decisão para avaliação dos resultados, representados por losangos; e, por fim, duas saídas, indicadas em vermelho — uma intermediária e outra final, gerada ao término do processo.

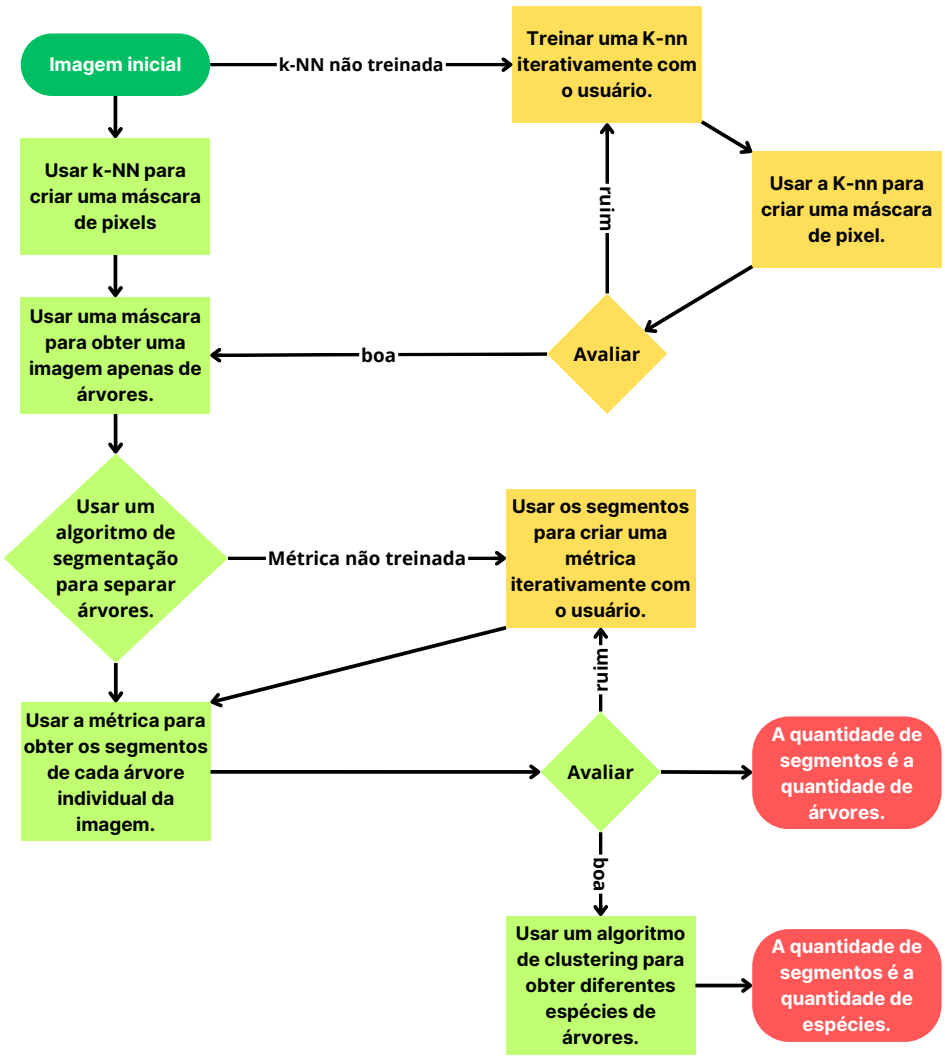
As etapas podem ser descritas da seguinte forma:

- **Imagem original**: Trata-se da imagem base utilizada para treinamento ou para ser submetida ao processo de classificação.
- **Treinar ou criar**: Nesta fase, o usuário realiza a seleção de amostras representativas da imagem base, com o objetivo de criar o classificador k-NN e definir a métrica de agrupamento. Para o k-NN, devem ser selecionados pontos diversos da cena, como

áreas de sombra, solo exposto e corpos d'água (referentes à classe NIO), bem como elementos característicos da vegetação, como folhas ou galhos com coloração distinta. A definição da métrica baseia-se na seleção de árvores com boa visibilidade visual, que são agrupadas manualmente para orientar o algoritmo de segmentação supervisionada.

- **Avaliar**: A avaliação é realizada de forma empírica, por inspeção visual dos resultados segmentados. Caso os objetos de interesse estejam corretamente identificados (por exemplo, se as árvores estão contidas na máscara ou bem separadas entre si), o processo pode avançar para a próxima etapa.
- **Saídas**: As saídas, representadas em vermelho, incluem o número de segmentos identificados após a segmentação, além de produtos intermediários gerados ao longo do fluxo.

Figura 4.1 – Fluxograma do sistema proposto



Fonte: Elaborado pelo autor

4.5 Extração de Indicadores Ecológicos

Com os dados obtidos a partir da segmentação e identificação das espécies nas imagens, é possível calcular indicadores de diversidade ecológica. Neste trabalho, foram utilizados os índices de Shannon e Simpson, citados no Capítulo 1, ambos amplamente aplicados para medir a riqueza e a uniformidade de espécies em uma determinada área.

O índice de Shannon (H') 1.5 estima a diversidade com base na incerteza da identificação aleatória de uma espécie (Shannon, 1948), enquanto o índice de Simpson (D) 1.4 mede a probabilidade de dois indivíduos pertencerem à mesma espécie (Simpson, 1949). Para facilitar a interpretação, utilizou-se $1 - D$, onde valores mais próximos de 1 indicam maior diversidade.

Como esses índices não possuem uma referência absoluta para julgar a qualidade ecológica, a interpretação exige comparação entre diferentes áreas. Assim, os valores obtido devem ser utilizados para indicar quais regiões apresentam maior diversidade e, portanto, maior potencial para preservação. Áreas com menor diversidade podem ser destinadas a outras finalidades, como manejo florestal ou recuperação ambiental.

4.6 Recursos Computacionais

O desenvolvimento e a execução dos experimentos foram realizados em um notebook da marca Lenovo com as especificações de hardware vistas na Tabela 4.1.

Tabela 4.1 – Especificações do ambiente computacional.

Componente	**Especificação**
Processador	Intel Core i5-1235U
Placa de Vídeo	Intel Iris Xe Graphics
Memória RAM	16 GB (2 x 8 GB Kingston Fury)
Armazenamento	512 GB NVMe SSD (SSSTC CL4-4D512-Q62)

Para a implementação do software, foram utilizadas as seguintes tecnologias: linguagem Python (v3.10.0) e sistema operacional Windows 11. Todas as bibliotecas e dependências do projeto estão listadas no arquivo `requirements.txt`, disponível no repositório Git mencionado no Capítulo 5.

5 Resultados

O código atual está disponível no GitHub, e pode ser acessado clicando em lekomatthias/ITS.

Todos os resultados deste Capítulo de resultados foram obtidos a partir dos recursos computacionais citados no Capítulo 4.6.

Ao todo, a base de teste possuia 12 fotografias de vista aérea tirada por drones. Sendo 2 de mais baixa resolução (HD) e 10 de alta resolução (4k). Para os testes demonstados neste capítulo foi escolhida uma imagem HD, pois esta possui 3 tipos de NIO's (grama, terra e asfalto), possui ao menos 13 árvores visíveis a olho nú (facilita a correção do treinamentos) e tem resultados rápidos para o feedback do treinamento de cada etapa.

5.1 Detecção de NIO's

Para o treinamento da KD-tree, foi utilizada uma imagem que apresenta o chão de terra com características bem definidas, conforme mostrado na Figura 5.1.

Figura 5.1 – Imagem aérea de drone base



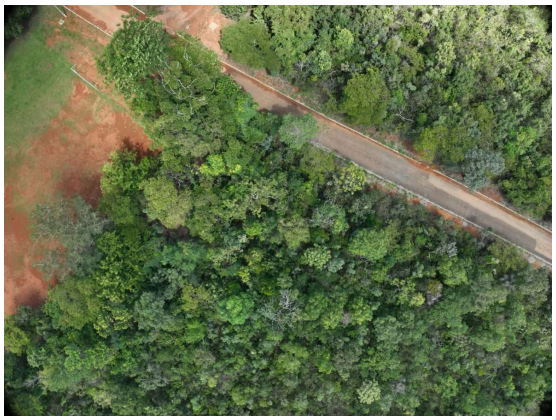
Fonte: Elaborada pelo autor

Com um treinamento iterativo, no qual o usuário indica o que deve ser ignorado e o que deve ser considerado, a máscara apresentada na Figura 5.2b é obtida. O treinamento é feito por meio de uma interface gráfica, onde o usuário deve selecionar regiões na imagem que possuem NIO's, que seria qualquer parte não pertencente à copa de alguma árvore e outra quantidade de áreas de interesse, sendo elas as copas. Para uma knn bem treinada, uma pequena quantidade de áreas é o suficiente (de 5 a 10 áreas de NIO's e de 5 a 10 de áreas de copas), para cada região um conjunto de pixels no entorno é adicionada ao espaço de amostras da knn. Um sistema mal treinado pode gerar uma pior separação posterior

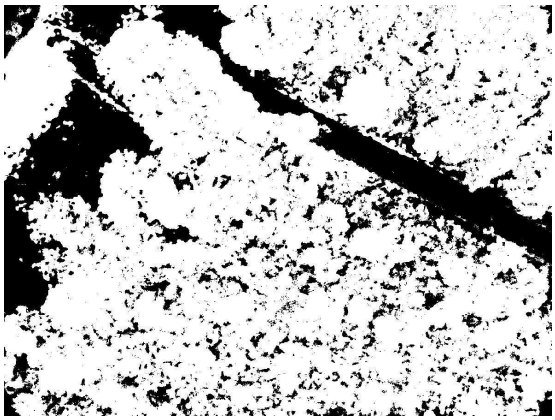
de árvores, pois ele pode classificar muitos pixels das copas como NIO's, sendo assim, é indicado que as regiões de exemplo sejam isoladas do exemplo oposto.

Figura 5.2 – Máscara da imagem de testes

(a) Base



(b) Máscara



Fonte: Elaborada pelo autor

5.2 Detecção de árvores

Foi aplicada a máscara e utilizado o SLIC do scikit-learn, assim foi obtida uma segmentação básica, onde apenas as árvores menores são segmentadas corretamente, enquanto as maiores podem estar separadas. Ao aplicar a matriz de métrica com Mahalanobis para os dados de distância física e de cor em todos os superpixels gerados, pôde ser observado que alguns deles são agrupados com base no *threshold* definido (este threshold possui variabilidade de caso a caso, então ele deve ser decidido empiricamente. No caso dos exemplos deste trabalho ele foi escolhido como 5,8). O resultado dessa etapa é ilustrado na Figura 5.3b.

Observação: Cada cor da imagem classificada significa um segmento diferente, e neste caso, um segmento deve ser uma árvore.

Ao contar uma árvore por segmento encontrado, infere-se que na imagem de testes existem 182 árvores.

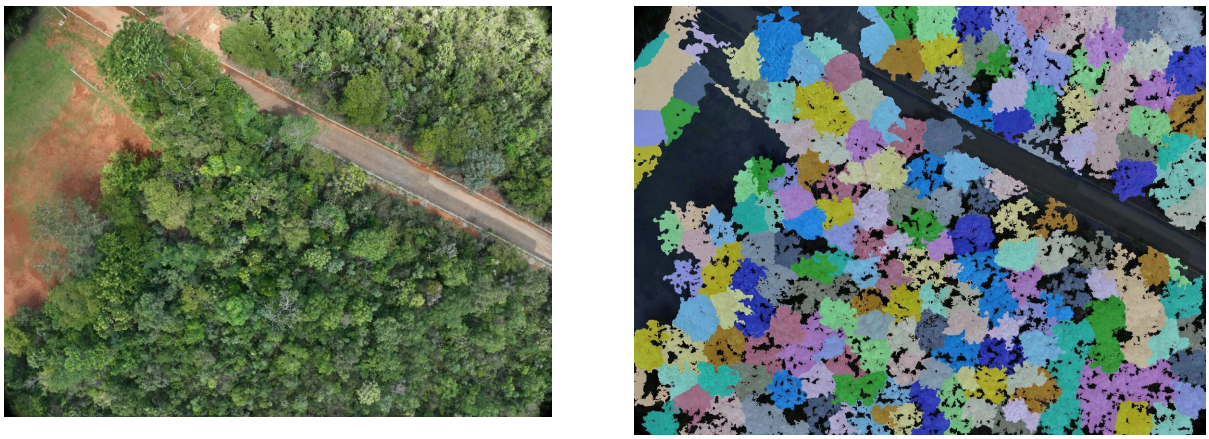
5.3 Detecção de espécies

Aplicando o K-means nos superpixels restantes, observa-se que a classificação pode gerar uma sequência de árvores classificadas como pertencentes à mesma espécie, como mostrado na Figura 5.4b.

Ao separar novamente apenas segmentos únicos, um total de 21 espécies foram encontradas. É um número razoável considerando a quantidade de árvores da etapa anterior,

Figura 5.3 – Imagem classificada com k-means

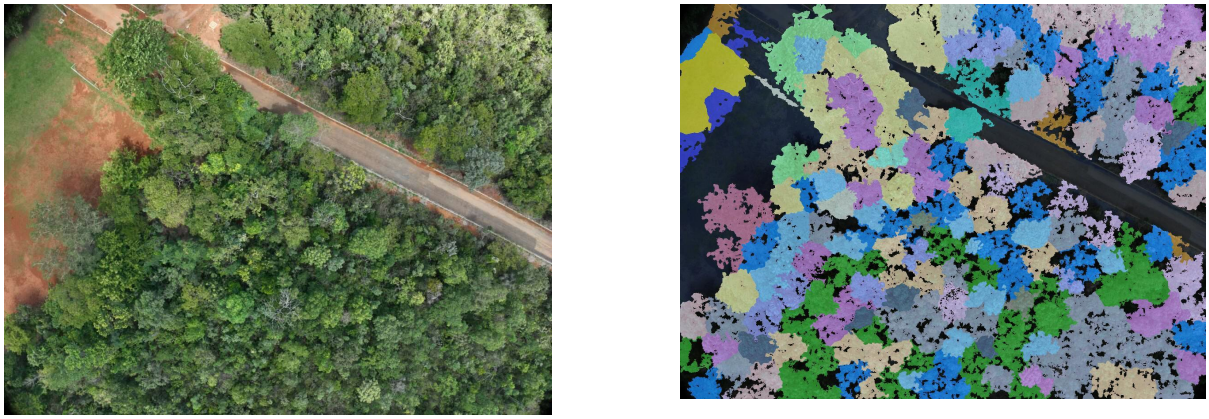
(a) Base (b) Classificação



Fonte: Elaborada pelo autor

Figura 5.4 – Imagem classificada com SLIC

(a) Base (b) Classificação

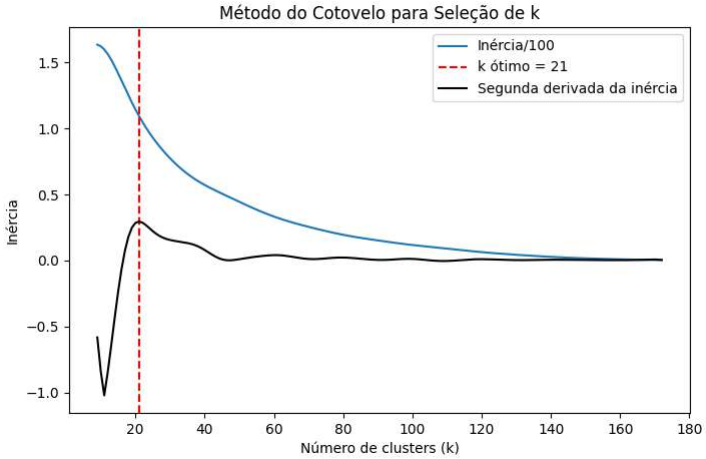


Fonte: Elaborada pelo autor

mas para o k-means, é preciso escolher a quantidade de cluster para se separação, ou seja, foi escolhido 21 como a quantidade de espécies, essa escolha é dada por um algoritmo de otimização. O algoritmo de otimização utilizado foi o método do "cotovelo", como é explicado a seguir.

Para o K-means, é necessário determinar o melhor valor de *k* para garantir uma divisão adequada. Para isso, pode-se aplicar o K-means em uma variedade de valores de *k* e analisar o desvio padrão entre os clusters gerados. A partir dessa análise, identificar o "cotovelo"na curva de clusters versus desvio gera o valor desejado, o que nos permitirá escolher o valor de *k* otimizado. A curva resultante é ilustrada na Figura 5.5.

Figura 5.5 – Gráfico número de clusters versus inércia



Fonte: Elaborada pelo autor

5.4 Comparação de resultados - segmentação

Seguindo os mesmos passos anteriores, foi testada uma boa diversidade de algoritmos.

Os segmentadores testados são:

- SLIC
- ETPS
- SEEDS
- ERS
- CRS

Como os algoritmos de segmentação foram feitos utilizando o docker citado no Capítulo 4, é preciso criar uma maneira de aplicar a máscara em imagens para que o algoritmo considere-as. A solução encontrada foi um pré-processamento onde a márcara multiplica a imagem inicial, tornando preta a área de NIO's. Depois da segmentação a máscara é aplicada novamente para garantir que apenas as árvores serão classificadas.

Os segmentos na Figura 5.6 são obtidos dessa maneira.

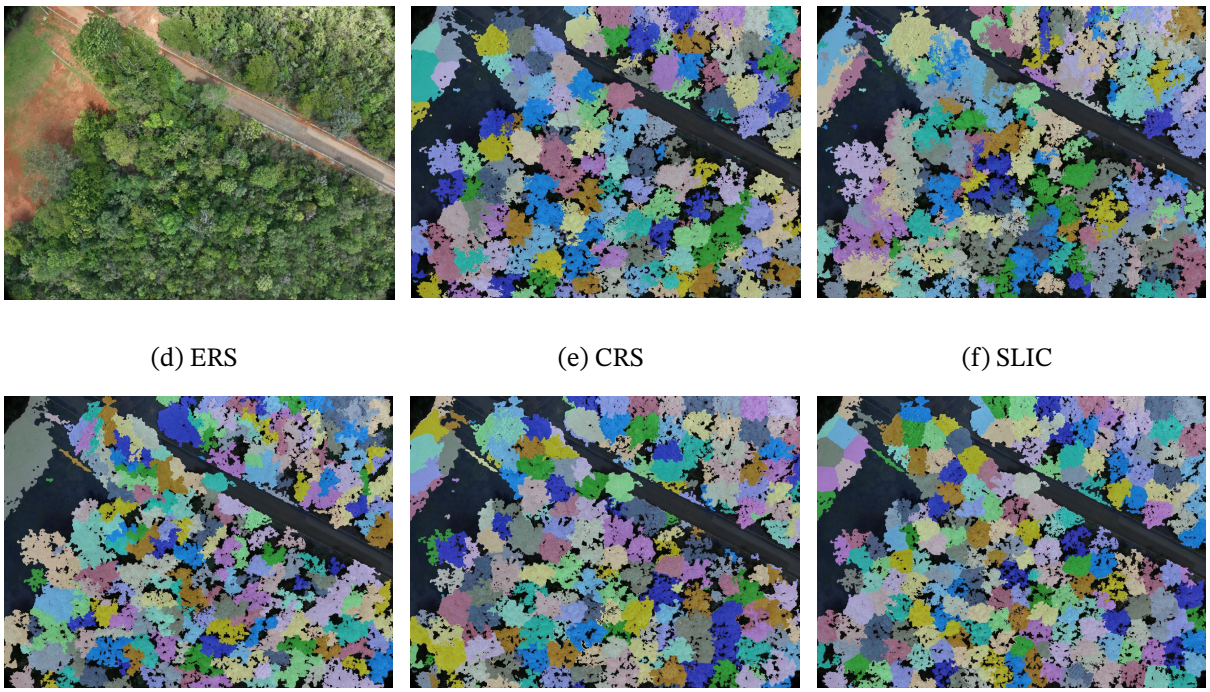
Todos os segmentos foram orientados a obter 200 segmentos iniciais, mas cada um adapta para uma quantidade diferente de árvores.

Para avaliar a segmentação, por falta de um ground truth, foram estabelecidas árvores (de maneira empírica) que tem uma separação clara para usar como base. Veja na Figura 5.7 as marcações que serão estabelecidas como base para avaliação.

As árvores marcadas na Figura 5.7 foram sempre levadas em consideração para ajuste de parâmetros e decisão sobre qual algoritmo enfatizar. Vale o destaque da grama, pois ela é muito difícil de classificar como não árvore pixel a pixel, então é desejado que ela seja considerada uma árvore e de uma espécie.

Figura 5.6 – Segmentos para comparação

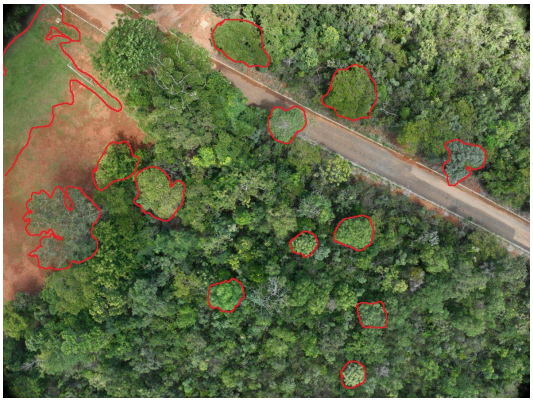
(a) Base (b) ETPS (c) SEEDS



(d) ERS (e) CRS (f) SLIC

Fonte: Elaborada pelo autor

Figura 5.7 – Imagem com ground truth



Fonte: Elaborada pelo autor

Para escolher entre um segmentador bom, foi comparada a Figura 5.7, e foram contabilizadas quantas das árvores marcadas não tiveram seus pixels agrupados a árvores vizinhas.

Analizando cada um dos segmentadores pode-se destacar que o SLIC 5.6f conseguiu um resultado bom, conseguindo separar bem uma árvore da outra, mas acaba super segmentando, o que pode não ser detectado pelo algoritmo feito para juntar segmentos que são da mesma árvore (9/13 árvores separadas sem juntar partes de outras árvores).

O ETPS 5.6b teve um resultado muito bom, ele destacou basicamente todas as marcações de 5.7, mas em quase todos os casos ele fez um segmento maior do que deveria, colocando

mais pixels indevidos como parte de uma árvore (9/13 árvores separadas sem juntar partes de outras árvores).

O SEEDS 5.6c não destacou nenhuma árvore muito bem, mas também não teve um resultado péssimo, ele junta muito bem partes semelhantes apesar de seguir muito bem bordas. Pode ser que um ajuste melhor de parâmetros possa trazer resultados melhores ou outros processamentos adicionais por exemplo (2/13 árvores separadas sem juntar partes de outras árvores).

O ERS 5.6d foi o único que conseguiu separar as gramíneas, sendo assim é evidente que para uma textura mais uniforme este foi o algoritmo que mais se destacou. Ele também conseguiu separar uma boa parte de outras árvores, mas apresentou a tendência de misturar uma árvore com outra, diminuindo muito sua qualdade na métrica escolhida (3/13 árvores separadas sem juntar partes de outras árvores).

O CRS 5.6e também apresentou uma boa segmentação, mas ainda assim super segmenta bastante, ele conseguiu separar muitas das árvores de teste, mas quando não consegue identificar uma árvore inteira ele segmenta em muitas áreas menores (7/13 árvores separadas sem juntar partes de outras árvores).

A Tabela 5.1 mostra a quantidade de árvores contadas para cada algoritmo.

Tabela 5.1 – Resultado dos segmentadores

segmentador	SLIC	ETPS	Seeds	ERS	CRS
árvores isoladas	9	9	2	3	7

5.5 Comparação de resultados - clusterização

Os seguintes algoritmos de clustering foram testados:

- K-Means (particional)
- Fuzzy C-Means (particional)
- DBSCAN (dencidade)
- OPTICS (dencidade)
- Mean-Shift (dencidade)
- Agglomerative Clustering (hierárquico)
- Birch (hierárquico)
- Affinity Propagation (redes neurais)
- SOM (redes neurais)
- Gaussian Mixture (baseado em modelo)

A fim de melhorar a leitura e visualização das imagens, elas serão postas em anexo ao final.

Para uma análise de cada tipo de clusterização, vamos observar os tipos de clusterização. E para observar qual é melhor ou pior, pode-se observar a proximidade das expécies classificasdas (árvores tendem a se ploriferar próximas uma das outras), a semelhança em tipo de folhas (podendo folhar ocuparem mais ou menos pixels dentro da imagem) e tamanho das copas separadas na etapa de contagem.

O primeiro tipo que é importante observar como problemático é a clusterização baseado em densidade. esse tipo de clusterização dificilmente consegue separar uma quantidade de espécies aceitável, tende a juntar quase todas em um tipo único, aqui testamos DBSCAN B.1, OPTICS D.1 e Mean-shift F.1.

Algoritmos hierárquicos têm uma tendência de separar muito bem cores destoantes, mas em contra partida, parece ser comum que eles cortem do ramo principal da copa a sua borda, a causa mais provável é a maior variedade entre cor e desvio padrão. Os hierárquicos testados foram Agglomerative clustering C.1 e Birch H.1.

Para clusterização baseada em redes neurais os resultados foram bons, mas ainda é preciso ajustes mais específicos para cada imagem, é o caso de affinity propagation E.1 e SOM J.1, nestes o maior problema é a constância, em algumas images testadas eles se saíam melhor, em outras pior, a depender dos segmentos utilizados.

Agora entramos nos resultados mais constantes, algoritmos de clusterização particional (k-means A.1 e fuzzy c-means I.1) e baseados em modelos (Gaussian mixture G.1). Já de início o fuzzy c-means se destaca como não adequado, pois ele consegue classificar uma árvore como ruído, e mesmo trabalhando com probabilidades ele apresenta resultados piores que seu semelhante k-means. O k-means apresenta resultados bem consistentes e bons, mas ainda não consegue conectar todas as árvores que pela visão humana deveriam ser consideradas da mesma espécie. O Gaussian mixture costuma ter bons resultados em diversos testes e em imagens diferentes mesmo que esporadicamente apresente resultados ruins como é o caso com sua segmentação com SEEDS em G.1.

Então, pelos testes apresentados, Gaussian mixture apresenta os melhores resultados preliminares, mas em contra partida tem de longe o maior tempo de processamente (pode ser vito na Tabela 5.2), chegando a ser 10 vezes mais lento que os outros algoritmos.

5.6 Índices de Diversidade

Com base nos resultados da segmentação e da classificação das espécies, foram calculados os índices de diversidade para a imagem de teste com a segmentação via SLIC (Figura 5.3) e clusterização K-Means (Figura 5.4). O índice de Simpson (na forma $1 - D$) apresentou valor de 0,89, enquanto o índice de Shannon foi de 3,69.

Ambos os resultados indicam uma diversidade significativa na área analisada. O valor de 0,89 no índice de Simpson sugere alta probabilidade de dois indivíduos aleatórios perten-

Tabela 5.2 – Tempo dos clusters

clusterizador	tempo (segundos)
K-Means	3.593
Fuzzy C-Means	7.931
DBSCAN	1.223
OPTICS	1.023
Mean-Shift	1.048
Agglomerative Clustering	1.219
BIRCH	3.097
Affinity Propagation	0.971
SOM	1.406
Gaussian Mixture	49.971

cerem a espécies distintas. Já o índice de Shannon reforça esse resultado, indicando uma distribuição relativamente equilibrada entre as espécies identificadas.

Esses indicadores permitem comparar a diversidade ecológica entre diferentes regiões, auxiliando na priorização de áreas para conservação ambiental ou outras finalidades estratégicas.

5.7 Parâmetros utilizados

Os parâmetros usados durante este trabalho têm o intuito de obter as melhores separações para imagem de base, que possui árvores bem definidas, área de NIO's e gramíneas. Como há uma necessidade inerente de intervensão humana em algumas etapas do processo, os parâmetros usados podem ser justificados com o que cada parâmetro muda na segmentação. Os parâmetros usados podem ser vistos na Tabela 5.3. A discussão específica de cada parâmetro é apresentada na Sessão 6.5.

Tabela 5.3 – Parâmetros dos segmentadores

Parâmetro	ETPS	Seeds	ERS	CRS	SLIC
Segmentos	200	200	200	200	200
Iterações	10	3	—	10	—
Sigma	—	—	5	—	1
Reg. Weight	0,1	—	—	—	—
Lambda	—	—	0,01	—	—
Compactness	—	—	—	5×10^{-4}	15
Bin	—	7	—	—	—

O que os parâmetros específicos como weight, lambda, compactness e bin significam são discutidos na Sessão 6.5.

6 Discussão

Os resultados apresentados revelam importantes aspectos sobre os desafios e as possibilidades na segmentação e classificação de árvores a partir de imagens aéreas obtidas por drones. Nesta seção, serão discutidos os principais achados, suas implicações e limitações.

6.1 Desempenho na detecção de NIOs

O método de geração de máscara utilizando KD-Tree mostrou-se eficiente na eliminação de regiões não relevantes (NIOs). A abordagem iterativa, com feedback do usuário, garantiu uma máscara precisa, o que foi fundamental para o desempenho das etapas subsequentes. Contudo, este processo ainda demanda intervenção manual, o que limita sua escalabilidade em aplicações de larga escala. Uma possível evolução seria automatizar esse processo utilizando modelos supervisionados ou redes neurais para detecção automática de NIOs.

6.2 Desempenho na segmentação

A análise dos diferentes algoritmos de segmentação evidencia que nenhum deles é universalmente superior. O algoritmo ETPS apresentou um excelente desempenho na separação de árvores, conseguindo capturar quase todas as referências do ground truth determinado. No entanto, ele tende a gerar segmentos maiores do que o ideal, o que pode comprometer etapas posteriores de clusterização.

Por outro lado, o SLIC foi eficaz na separação individual de árvores, mas sofre de supersegmentação. Isso torna indispensável o uso de uma etapa posterior de agrupamento de superpixels, como feito com a métrica de Mahalanobis.

O destaque do ERS na separação de gramíneas indica que sua sensibilidade a texturas mais homogêneas pode ser aproveitada em contextos específicos, como a separação de sub-bosques, vegetação rasteira ou até mesmo rios e/ou lagos.

Por fim, algoritmos como SEEDS e CRS apresentaram desempenho intermediário, sendo sensíveis tanto ao ajuste de parâmetros quanto à variação na textura e cor das copas.

6.3 Desempenho na clusterização

A etapa de clusterização revelou que algoritmos baseados em densidade, como DBSCAN, OPTICS e Mean-Shift, não são adequados para este tipo de dado. Isso se deve à

dificuldade desses métodos em lidar com a variação natural da densidade das copas e das características cromáticas das árvores.

Os métodos hierárquicos (Agglomerative e Birch) apresentaram uma divisão satisfatória quando as árvores possuem cores muito distintas. Contudo, mostraram fragilidade na manutenção da integridade espacial das copas, frequentemente dividindo partes que deveriam pertencer a uma mesma árvore.

Entre os métodos avaliados, os algoritmos particionais (K-Means e Fuzzy C-Means) e os baseados em modelos (Gaussian Mixture) mostraram-se mais robustos. Entretanto, o Fuzzy C-Means apresentou desempenho inferior, frequentemente classificando árvores como ruído, o que demonstra sua menor adequação ao problema apresentado.

O K-Means manteve bons resultados, com uma classificação consistente e estável. Contudo, o K-Means apresenta uma desvantagem típica: a necessidade de definir previamente o número de clusters. Essa escolha pode ser subjetiva e depende de um bom método de otimização, como o método do cotovelo que foi utilizado.

O Gaussian Mixture apresentou o melhor desempenho qualitativo, com alta capacidade de adaptação às variações nas cores e texturas das árvores. Seu maior entrave, porém, é o custo computacional elevado — cerca de 10 vezes maior que os demais —, o que o torna pouco prático para aplicações em larga escala.

Os algoritmos baseados em redes neurais, como SOM e Affinity Propagation, mostraram potencial, porém ainda instáveis, com resultados fortemente dependentes da configuração inicial e da imagem aplicada.

6.4 Implicações dos resultados

Os resultados demonstram que a combinação de uma segmentação adequada (como ETPS ou SLIC) com um método robusto de clusterização (como Gaussian Mixture ou K-Means) é uma estratégia promissora para a detecção e classificação de árvores em imagens aéreas.

Além disso, ficou evidente que nenhuma abordagem isolada é suficiente para lidar com a variabilidade natural dos ambientes florestais. A presença de sombras, variação nas cores das copas, sobreposição de ramos e vegetação rasteira são fatores que impactam diretamente na performance dos algoritmos.

Dado que pode-se aplicar o índice de Simpson ou de Shannon, é viável utilizar o método proposto para saber qual área é mais adequada para ser preservada.

6.5 Parâmetros

Os parâmetros detalhados na Tabela 5.3 foram definidos inicialmente com base na Figura 5.1 e, posteriormente, avalidados em 11 outras imagens de teste.

A única entrada comum a todos os algoritmos é o parâmetro "segmentos". Este valor representa uma estimativa inicial da quantidade de segmentos desejados na imagem. Embora o número final de segmentos raramente coincida com essa estimativa inicial, ele pode ser ajustado conforme a área de NIOs e a altura de captura da imagem pelo drone. Para reduzir a subjetividade dessa entrada, um algoritmo com interface gráfica (GUI) foi desenvolvido. Essa ferramenta permite ao usuário visualizar o tamanho das árvores na tela e, com base na dimensão considerada mais adequada, um valor inicial para o parâmetro "segmentos"é calculado automaticamente.

O número de interações foi determinado considerando o equilíbrio entre o tempo de processamento e a precisão da aderência às bordas das copas das árvores.

O parâmetro sigma geralmente promove uma suavização das bordas. No entanto, como demonstrado na Figura 5.5, essa suavização não compromete a qualidade do algoritmo ERS.

Parâmetros como regularization weight, lambda, compactness e bin são análogos, embora específicos para algoritmos distintos. Todos eles têm como objetivo controlar a compactação dos segmentos. Cada algoritmo processa esses parâmetros de forma diferente, resultando em respostas variadas. Notavelmente, o algoritmo Seeds produz uma resposta mais granularizada com valores de "bin"maiores, mas isso acarreta um consumo significativo de memória. Um segmento mais compacto tende a apresentar bordas menos rugosas, o que pode levar a uma segmentação mais arredondada e, consequentemente, menos precisa em termos de detalhe.

6.6 Limitações

Entre as principais limitações do presente trabalho, destacam-se:

- A ausência de um ground truth robusto e quantitativo, o que dificulta a validação objetiva dos resultados;
- A dependência de parâmetros subjetivos, tanto na segmentação quanto na clusterização e
- Alto custo computacional de alguns métodos, especialmente Gaussian Mixture.

7 Conclusão

Este trabalho propôs um método para contagem e identificação individual de árvores em imagens aéreas obtidas por drones, com foco na extração automatizada de informações relevantes para aplicações ambientais, como monitoramento de ecossistemas e manejo sustentável.

A abordagem desenvolvida baseia-se em uma combinação de técnicas de aprendizado supervisionado (k-NN), segmentação de superpixels e algoritmos de clusterização não supervisionada, visando primeiro separar vegetação de objetos não interessantes (NIOs), depois isolar cada copa de árvore e, por fim, agrupá-las por similaridade espectral, inferindo possíveis espécies.

Os testes demonstraram que:

- A segmentação via Extended Topology Preserving Segmentation se mostrou eficaz na separação de copas em áreas com alta densidade arbórea, embora com tendência a supersegmentar;
- O Simple Linear Iterative Clustering, embora produza resultados com superpixels bem definidos, exige uma boa métrica para juntar segmentos e representar árvores de forma precisa;
- Dentre os algoritmos de clusterização, o Gaussian Mixture obteve os melhores resultados em termos de separação espectral, embora com elevado custo computacional. O k-means, por sua vez, apresentou desempenho estável e aceitável com menor custo.

O método desenvolvido provou-se funcional mesmo com imagens sem rótulos e em contextos onde há variabilidade significativa de coloração, sobreposição de copas e ausência de dados topográficos.

Contudo, limitações importantes foram identificadas, como a necessidade de intervenção humana em fases de treinamento, a ausência de validação quantitativa por ground truth confiável, e a dependência de parâmetros empíricos, especialmente nos processos de clusterização.

Em resumo, este trabalho oferece uma base sólida e replicável para futuros estudos voltados à análise automatizada de florestas por sensoriamento remoto. Apesar das limitações, os resultados obtidos indicam o potencial da abordagem proposta, com espaço evidente para aprimoramentos em automação, desempenho e precisão.

8 Propostas para Estudo Futuro

Todos os resultados obtidos podem ser utilizados para continuar a pesquisa sobre contagem individual de árvores, sendo assim, pode-se considerar as seguintes melhorias para otimizar tempo e resultados:

8.1 Melhorias possíveis para o mascaramento

- Uma análise estatística como esparcidade de blocos também pode revelar um bom resultado. Uma análise em cima da imagem crua sem uma surpervisão humana pode aumentar muito a chance de erro consequênte da alta variabilidade de árvores, sendo assim, um tratamento híbrido entre cor e variabilidade pode ser um caminho viável.

8.2 Melhorias possíveis para segmentação

- Testar uma outra forma de juntar os segmentos pode melhorar muito a eficiência do SLIC, suprindo o ponto negativo de supersegmentação.
- Ajustar melhor os segmentos do ETPS pode gerar um resultado mais preciso, melhorando junto com a métrica apresentada pode fazer com que o ETPS faça segmentos menores, isolando melhor cada árvore.

8.3 Melhorias possíveis para clusterização

- Testar novos algoritmos pode ser a melhor alternativa, os melhores resultados vieram de algoritmos clássicos (k-means e gaussian mixture), então variações modernas podem gerar uma clusterização mais precisa e eficiente do que as testadas (implementações padrão da biblioteca sklearn do python).
- O resultado da clusterização depende muito dos dados utilizados no algoritmo, sendo assim, encontrar uma métrica diferente para caracterizar cada superpixel pode trazer resultados novos bem diferentes dos apresentados neste trabalho.

Referências

ACHANTA, R.; SHAJI, A.; SMITH, K.; LUCCHI, A.; FUA, P.; SÜSSTRUNK, S. Slic superpixels compared to state-of-the-art superpixel methods. *In*: **IEEE Transactions on Pattern Analysis and Machine Intelligence**. [*S.l.*]: IEEE, 2012. v. 34, n. 11, p. 2274–2282. Citado na p. 16.

ANKERST, M.; BREUNIG, M. M.; KRIEGEL, H.-P.; SANDER, J. Optics: Ordering points to identify the clustering structure. **SIGMOD Record**, ACM, v. 28, n. 2, p. 49–60, 1999. Citado na p. 14.

BENTLEY, J. L. Multidimensional binary search trees used for associative searching. **Communications of the ACM**, ACM, v. 18, n. 9, p. 509–517, 1975. Citado na p. 21.

BERGH, M. Van den; BOIX, X.; ROIG, G.; CAPITANI, B. de; GOOL, L. V. Seeds: Superpixels extracted via energy-driven sampling. *In*: **Proceedings of the European Conference on Computer Vision**. [*S.l.*]: Springer, 2012. v. 7578, p. 13–26. DOI 10.1007/978-3-642-33786-4$_2$. Citado na p. 16.

CHENG, Y. Mean shift, mode seeking, and clustering. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 17, n. 8, p. 790–799, 1995. Citado na p. 15.

DAVé, R.; KRISHNAPURAM, R. Robust clustering methods: A unified view. **IEEE Transactions on Fuzzy Systems**, IEEE, v. 5, n. 2, p. 270–293, 1997. Citado na p. 15.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *In*: AAAI PRESS. **Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)**. [*S.l.*], 1996. p. 226–231. Citado na p. 14.

FIX, E.; HODGES, J. L. **Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties**. 1951. USAF School of Aviation Medicine, Randolph Field, Texas. Citado na p. 17.

FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. Sparse inverse covariance estimation with the graphical lasso. **arXiv preprint arXiv:0805.1096**, 2008. DOI 10.48550/arXiv.0805.1096. Citado na p. 14.

KOHONEN, T. **Self-Organizing Maps**. 3rd. ed. [*S.l.*]: Springer, 1995. v. 30. (Springer Series in Information Sciences, v. 30). Citado na p. 15.

LIU, M.-Y.; TUZEL, O.; RAMALINGAM, S.; CHELLAPPA, R. Entropy rate superpixel segmentation. *In*: **Proceedings of the IEEE Conference on Computer Vision and Pat-**

tern Recognition. [*S.l.*]: IEEE, 2011. p. 2097–2104. DOI 10.1109/CVPR.2011.5995323. Citado na p. 17.

MACQUEEN, J. B. Some methods for classification and analysis of multivariate observations. **Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability**, University of California Press, v. 1, p. 281–297, 1967. Citado na p. 13.

MESTER, R.; CONRAD, C.; GUEVARA, A. Multichannel segmentation using contour relaxation: Fast super-pixels and temporal propagation. *In*: **Proceedings of the 17th Scandinavian Conference on Image Analysis**. [*S.l.*]: Springer, 2011. v. 6688, p. 250–261. DOI 10.1007/978-3-642-21227-$7_2$4. Citado na p. 17.

MÜLLNER, D. Modern hierarchical, agglomerative clustering algorithms. **arXiv preprint arXiv:1109.2378**, 2011. Citado na p. 14.

SHANNON, C. E. A mathematical theory of communication. **Bell System Technical Journal**, v. 27, n. 3, p. 379–423, 1948. Citado nas pp. 18 e 24.

SIMPSON, E. Measurement of diversity. **Nature**, v. 163, p. 688, 1949. Citado nas pp. 18 e 24.

STUTZ, D.; HERMANS, A.; LEIBE, B. Superpixels: An evaluation of the state-of-the-art. **Computer Vision and Image Understanding**, Elsevier, v. 166, p. 1–27, 2018. DOI 10.1016/j.cviu.2017.05.006. Citado na p. 22.

WANG, Y.; YANG, X.; ZHANG, L.; FAN, X.; YE, Q.; FU, L. Individual tree segmentation and tree-counting using supervised clustering. **Computers and Electronics in Agriculture**, Elsevier, v. 205, p. 107629, 2023. DOI 10.1016/j.compag.2023.107629. Citado nas pp. 18, 21 e 22.

XU, R.; II, D. W. Survey of clustering algorithms. **IEEE Transactions on Neural Networks**, IEEE, v. 16, n. 3, p. 645–678, 2005. DOI 10.1109/TNN.2005.845141. Citado na p. 22.

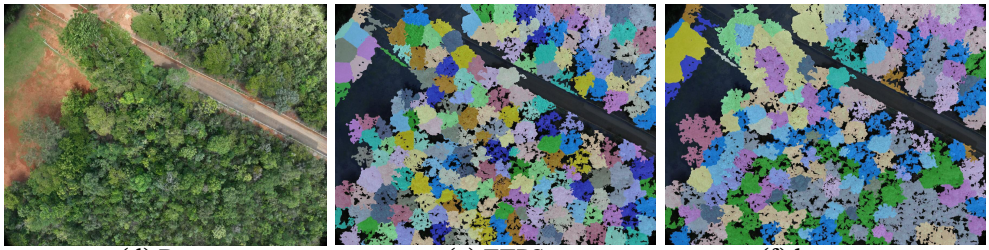
YAO, J.; BOBEN, M.; FIDLER, S.; URTASUN, R. Real-time coarse-to-fine topologically preserving segmentation. *In*: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [*S.l.*]: IEEE, 2015. p. 2947–2955. DOI 10.1109/CVPR.2015.7298913. Citado na p. 16.

ZHANG, T.; RAMAKRISHNAN, R.; LIVNY, M. Birch: An efficient data clustering method for very large databases. *In*: ACM. **Proceedings of the ACM SIGMOD Conference on Management of Data**. [*S.l.*], 1996. p. 103–114. Citado na p. 15.

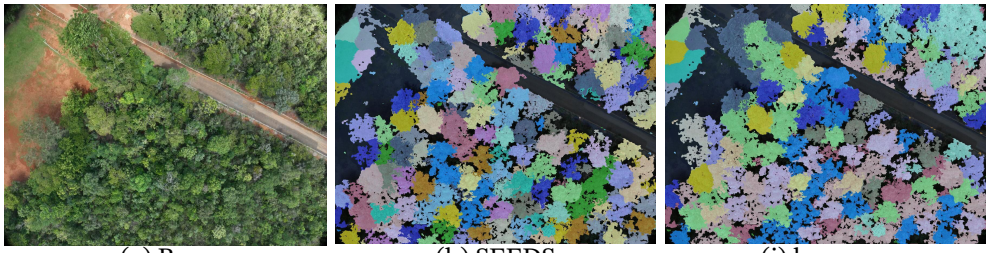
Apêndice A – k-means

Figura A.1 – Segmentos com cada segmentador clusterizados com k-means

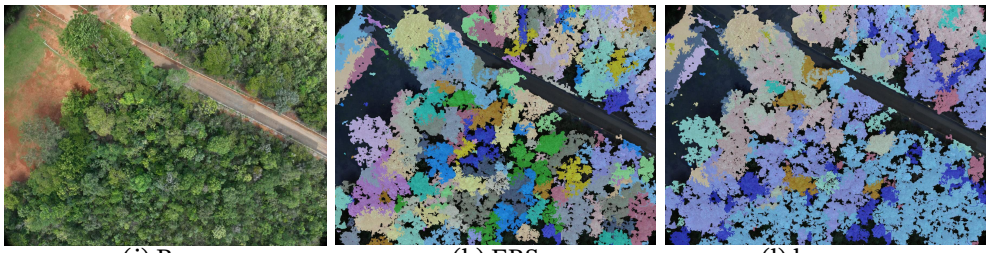
(a) Base (b) SLIC (c) k-means



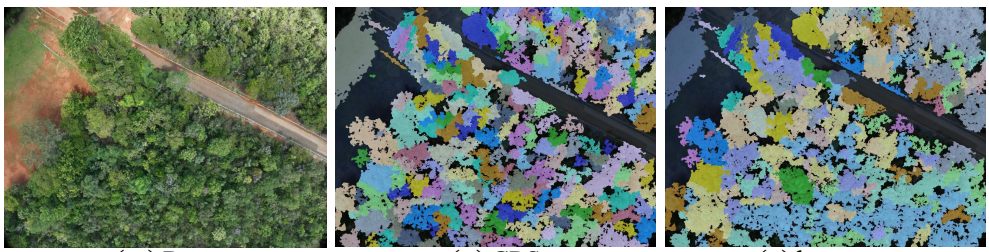
(d) Base (e) ETPS (f) k-means



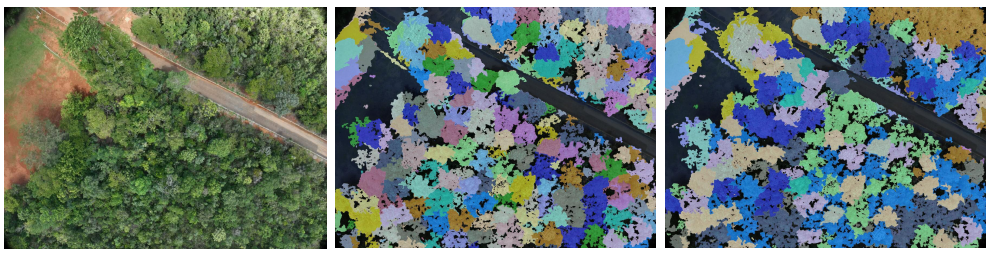
(g) Base (h) SEEDS (i) k-means



(j) Base (k) ERS (l) k-means



(m) Base (n) CRS (o) k-means

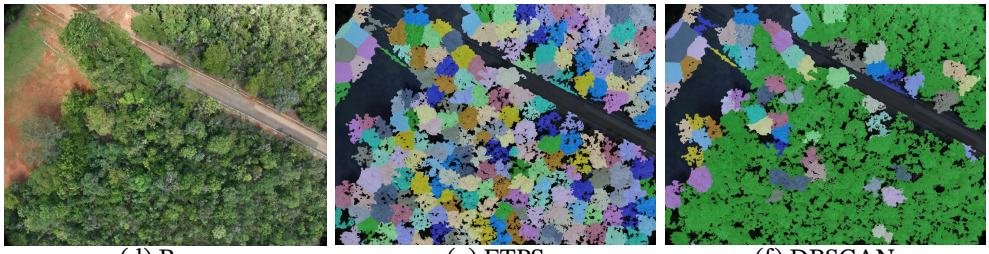


Fonte: Elaborada pelo autor

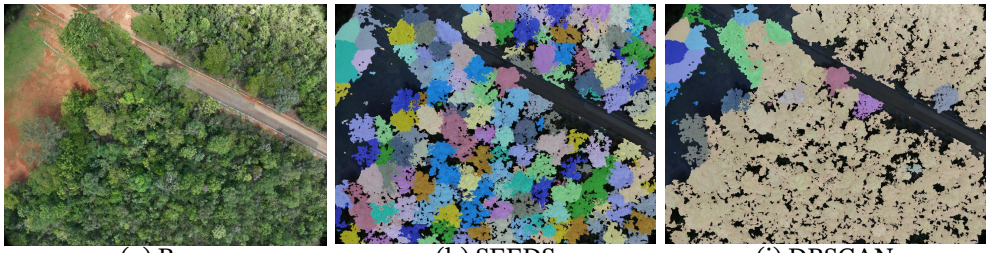
Apêndice B – DBSCAN

Figura B.1 – Segmentos com cada segmentador clusterizados com DBSCAN

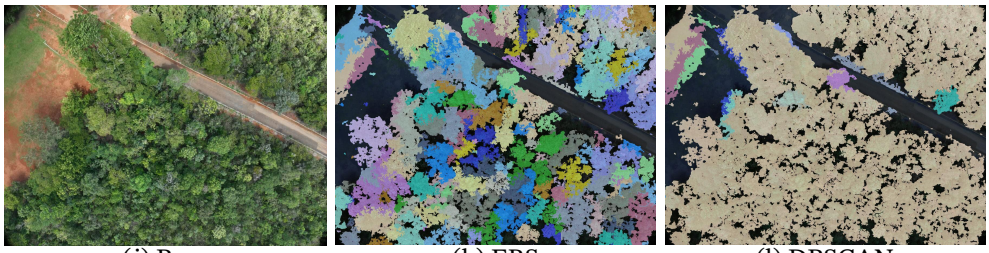
(a) Base (b) SLIC (c) DBSCAN



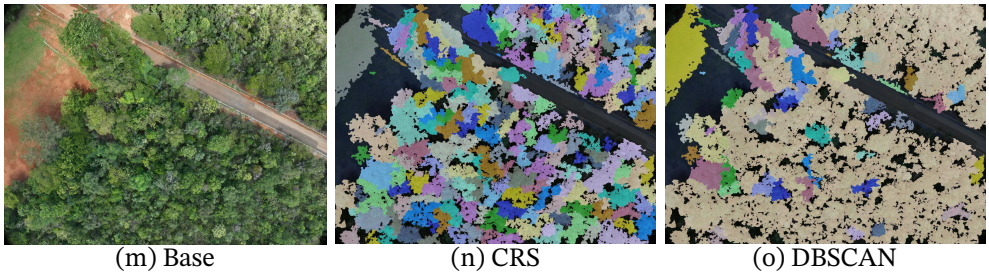
(d) Base (e) ETPS (f) DBSCAN



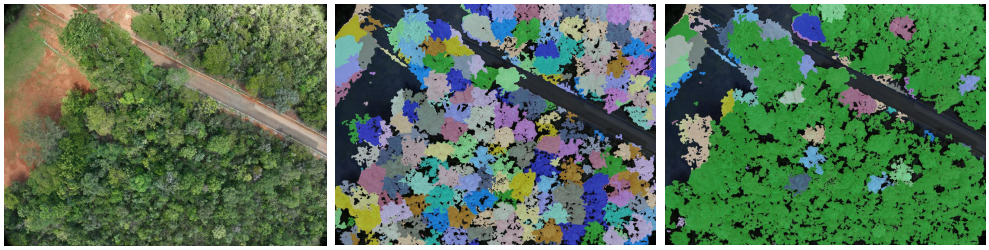
(g) Base (h) SEEDS (i) DBSCAN



(j) Base (k) ERS (l) DBSCAN



(m) Base (n) CRS (o) DBSCAN

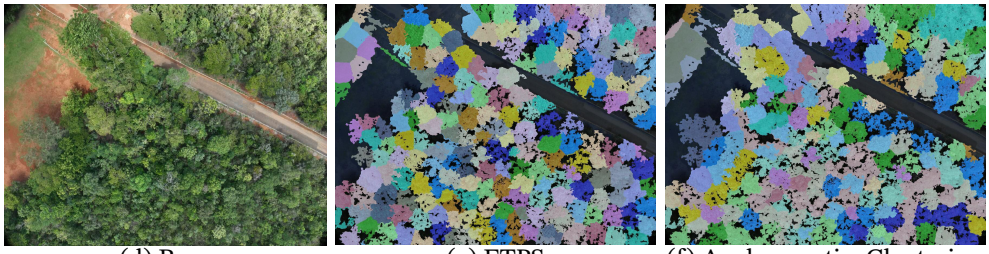


Fonte: Elaborada pelo autor

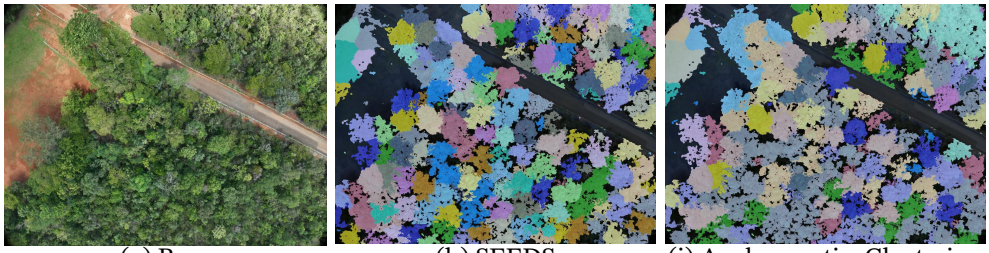
Apêndice C – Agglomerative Clustering

Figura C.1 – Segmentos com cada segmentador clusterizados com AgglomerativeClustering

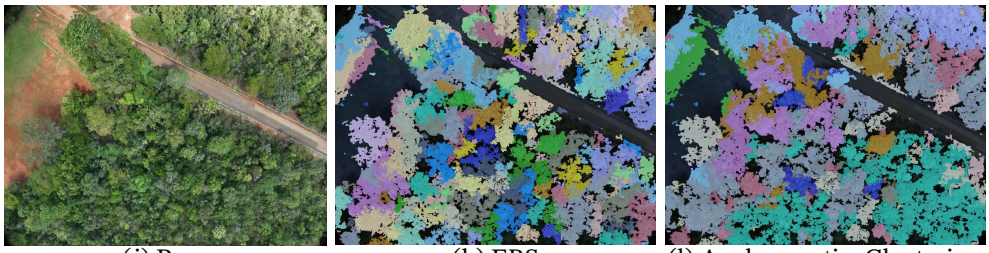
(a) Base (b) SLIC (c) AgglomerativeClustering



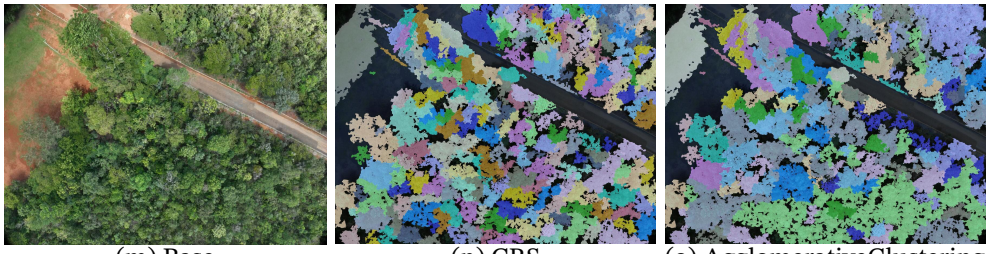
(d) Base (e) ETPS (f) AgglomerativeClustering



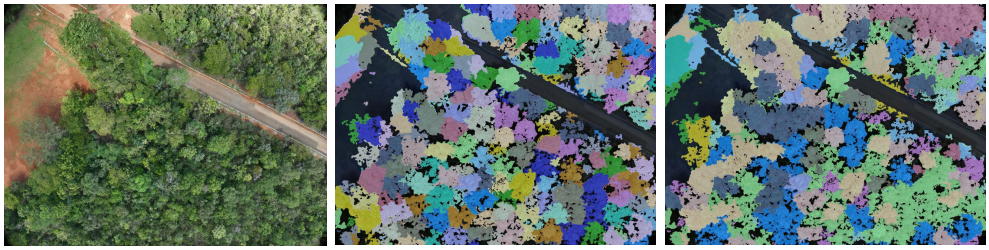
(g) Base (h) SEEDS (i) AgglomerativeClustering



(j) Base (k) ERS (l) AgglomerativeClustering



(m) Base (n) CRS (o) AgglomerativeClustering

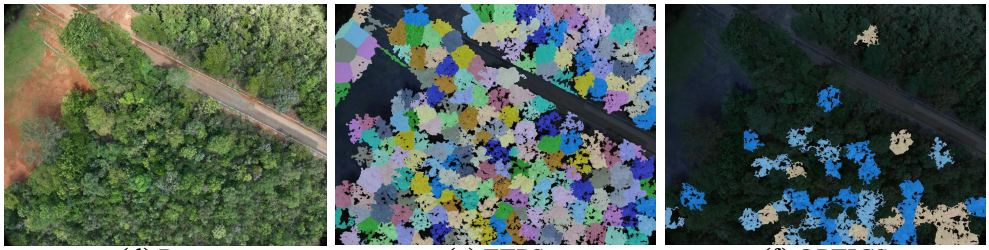


Fonte: Elaborada pelo autor

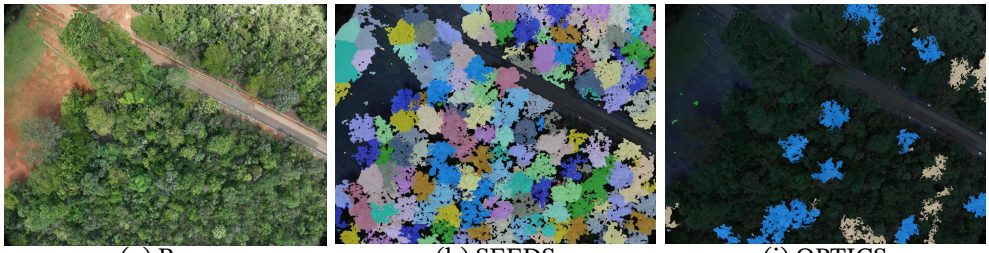
Apêndice D – OPTICS

Figura D.1 – Segmentos com cada segmentador clusterizados com OPTICS

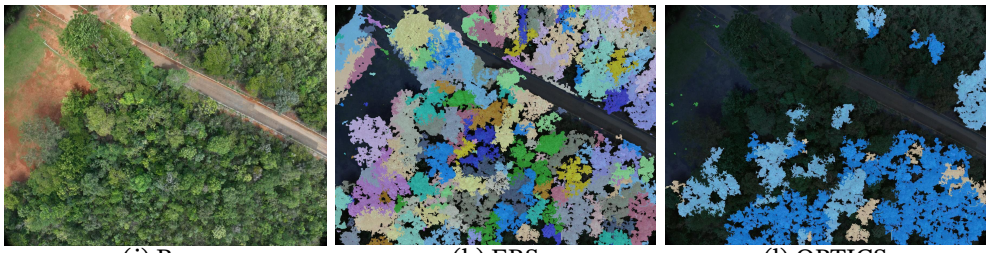
(a) Base (b) SLIC (c) OPTICS



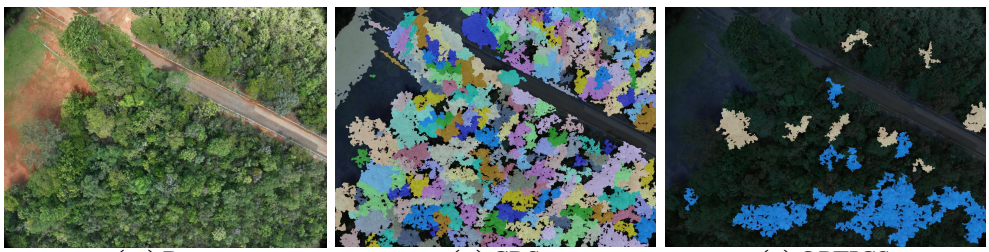
(d) Base (e) ETPS (f) OPTICS



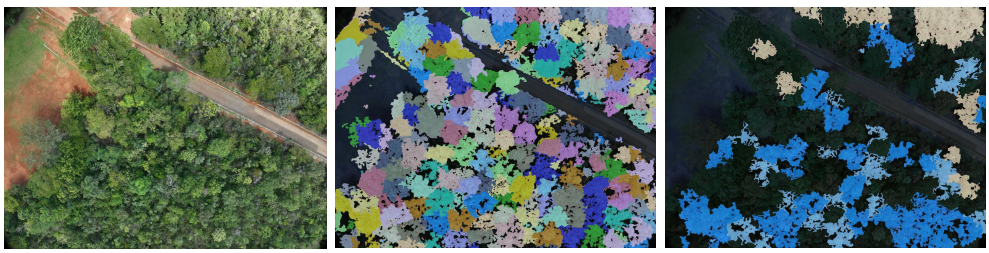
(g) Base (h) SEEDS (i) OPTICS



(j) Base (k) ERS (l) OPTICS



(m) Base (n) CRS (o) OPTICS

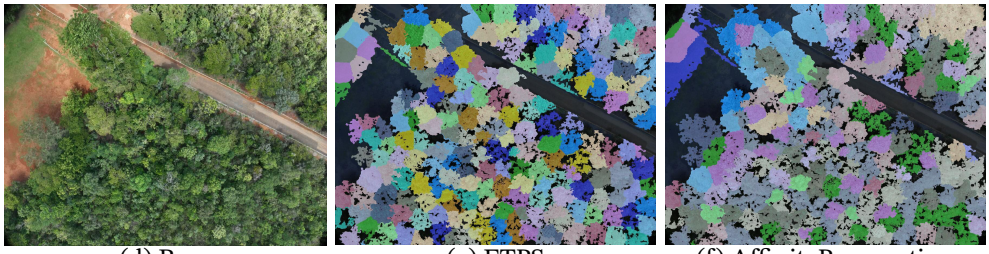


Fonte: Elaborada pelo autor

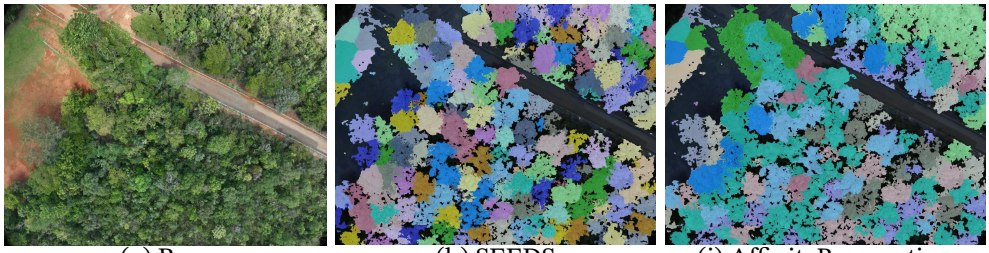
Apêndice E – Affinity Propagation

Figura E.1 – Segmentos com cada segmentador clusterizados com AffinityPropagation

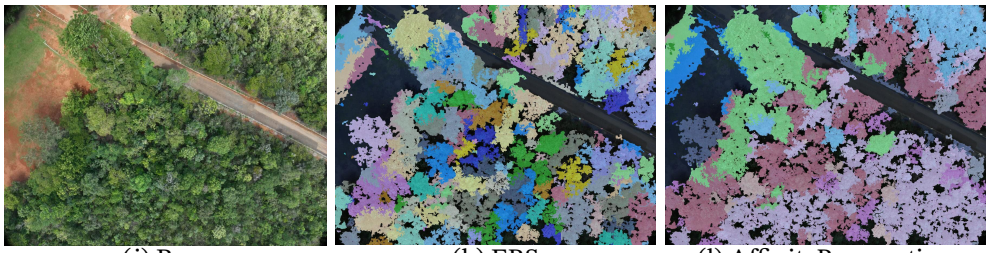
(a) Base (b) SLIC (c) AffinityPropagation



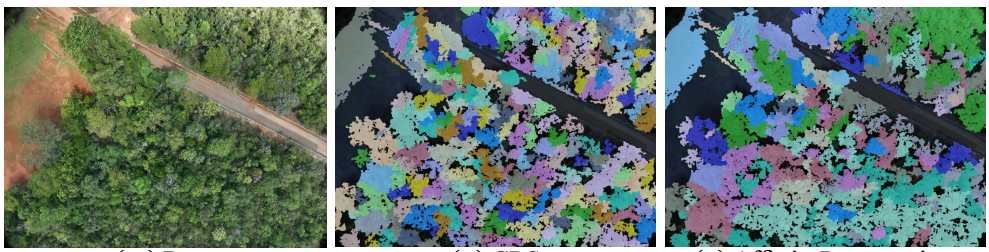
(d) Base (e) ETPS (f) AffinityPropagation



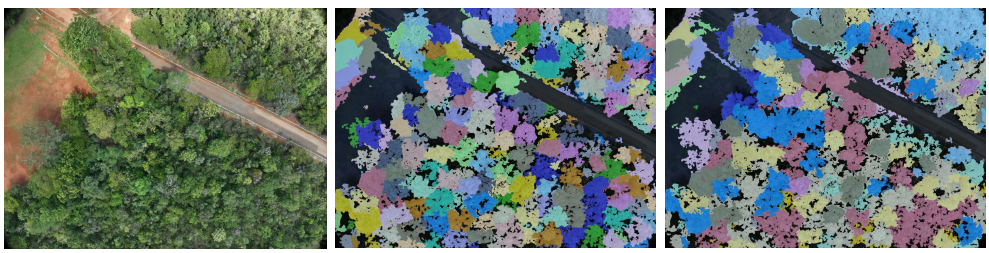
(g) Base (h) SEEDS (i) AffinityPropagation



(j) Base (k) ERS (l) AffinityPropagation



(m) Base (n) CRS (o) AffinityPropagation

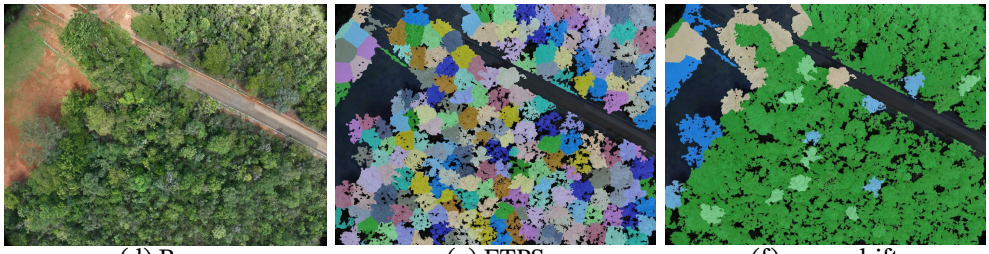


Fonte: Elaborada pelo autor

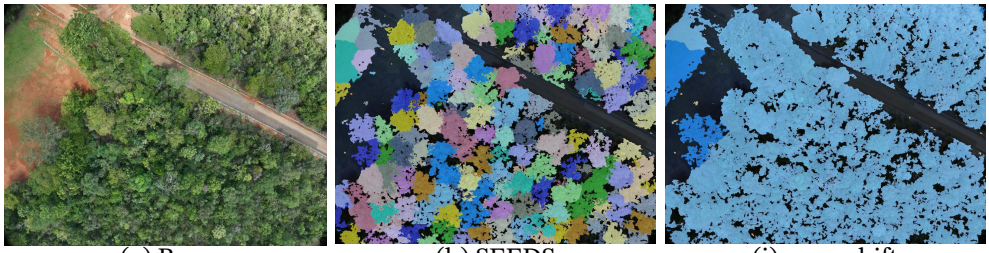
Apêndice F – Mean-Shift

Figura F.1 – Segmentos com cada segmentador clusterizados com meanshift

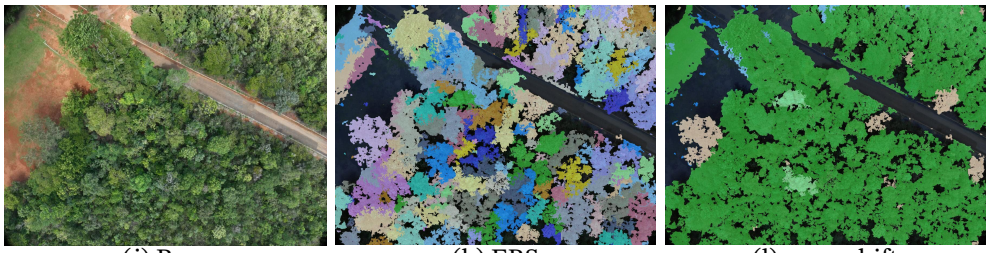
(a) Base (b) SLIC (c) meanshift



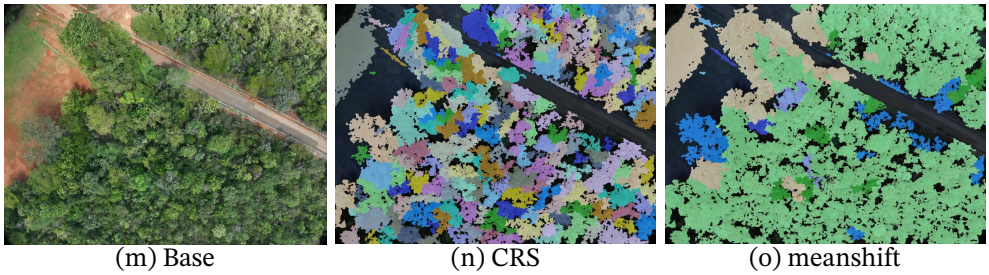
(d) Base (e) ETPS (f) meanshift



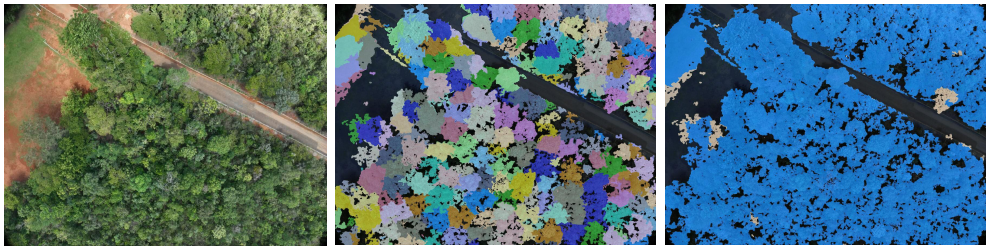
(g) Base (h) SEEDS (i) meanshift



(j) Base (k) ERS (l) meanshift



(m) Base (n) CRS (o) meanshift

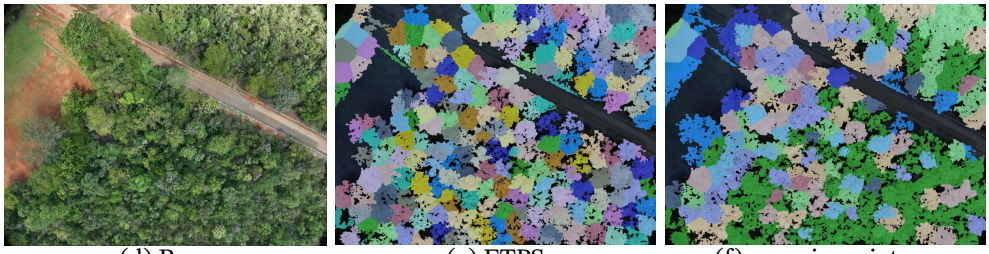


Fonte: Elaborada pelo autor

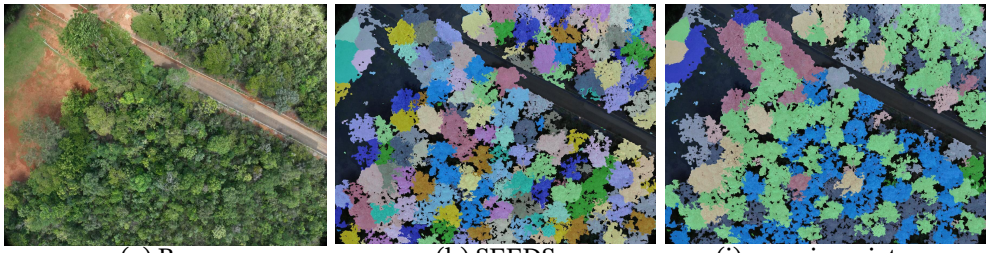
Apêndice G – Gaussian Mixture

Figura G.1 – Segmentos com cada segmentador clusterizados com gaussianmixture

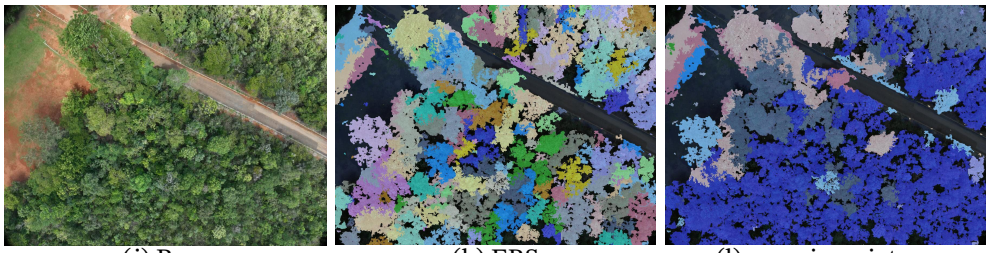
(a) Base (b) SLIC (c) gaussianmixture



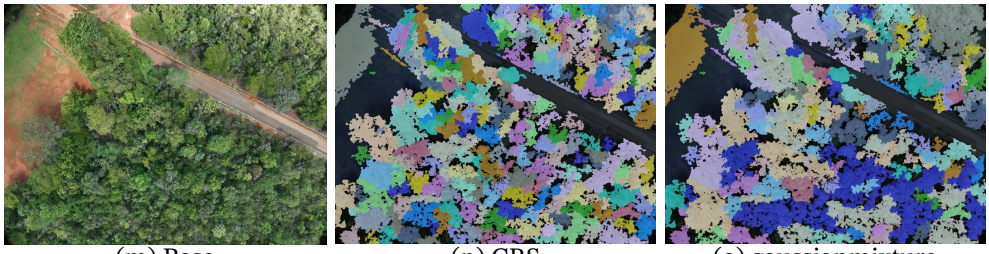
(d) Base (e) ETPS (f) gaussianmixture



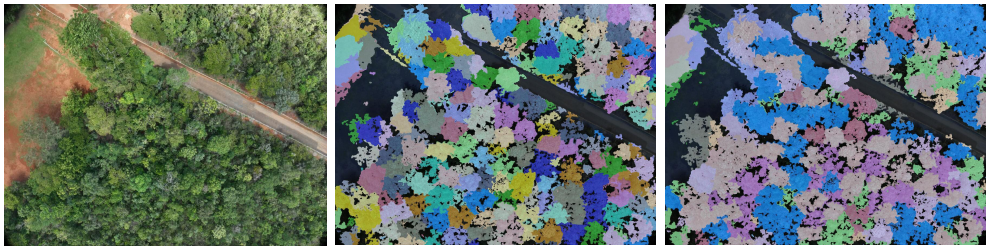
(g) Base (h) SEEDS (i) gaussianmixture



(j) Base (k) ERS (l) gaussianmixture



(m) Base (n) CRS (o) gaussianmixture

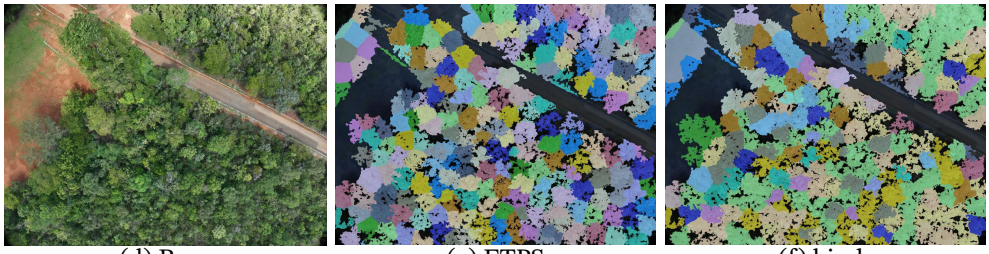


Fonte: Elaborada pelo autor

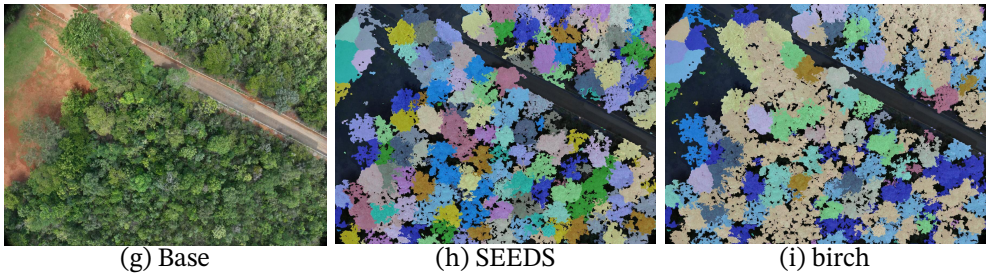
Apêndice H – Birch

Figura H.1 – Segmentos com cada segmentador clusterizados com birch

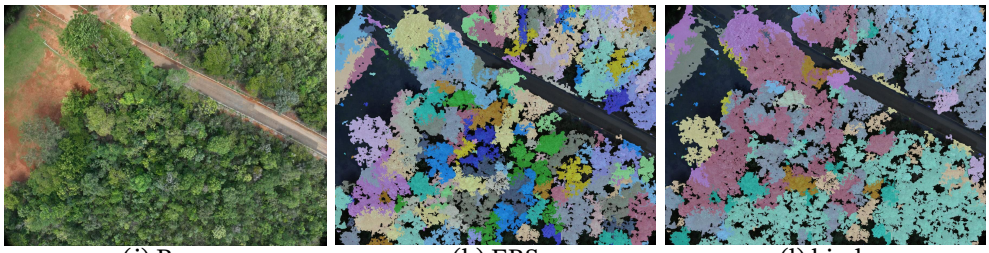
(a) Base (b) SLIC (c) birch



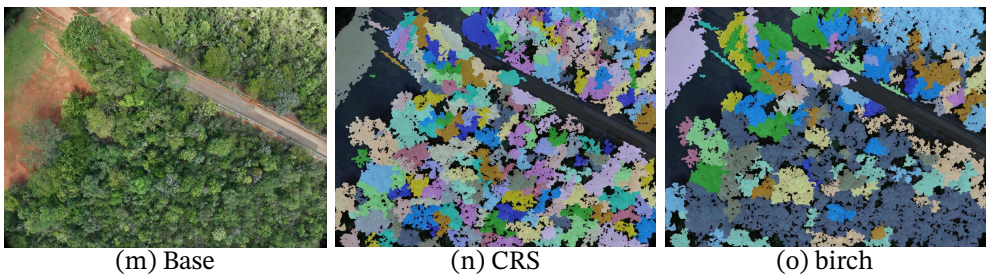
(d) Base (e) ETPS (f) birch



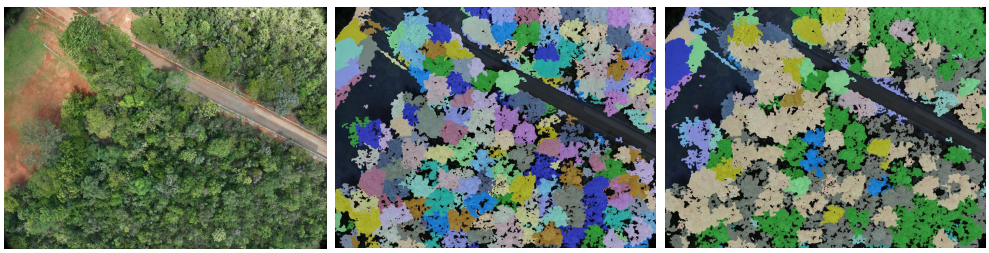
(g) Base (h) SEEDS (i) birch



(j) Base (k) ERS (l) birch



(m) Base (n) CRS (o) birch

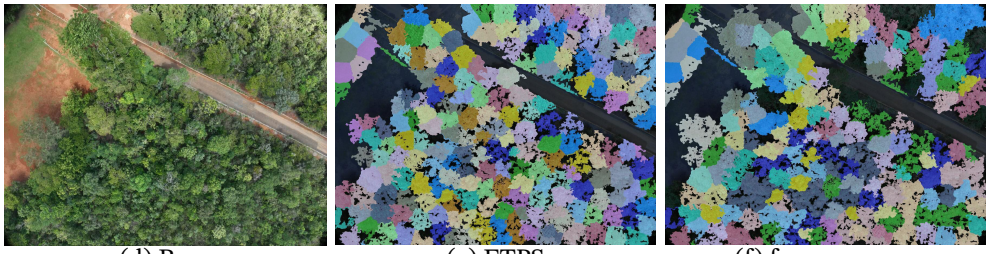


Fonte: Elaborada pelo autor

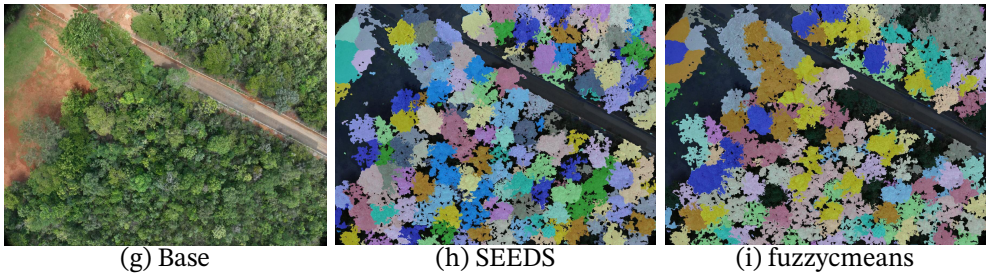
Apêndice I – Fuzzy C-Means

Figura I.1 – Segmentos com cada segmentador clusterizados com fuzzycmeans

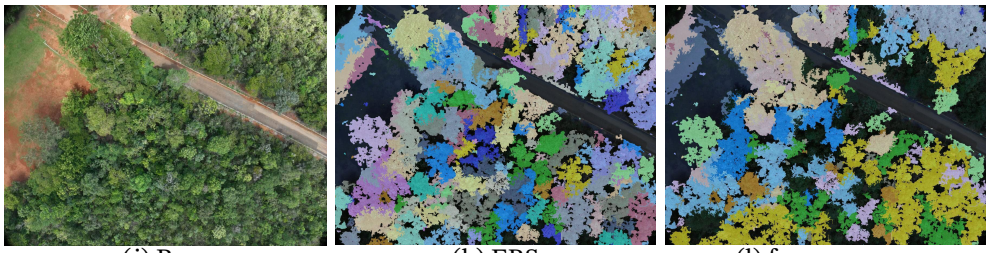
(a) Base (b) SLIC (c) fuzzycmeans



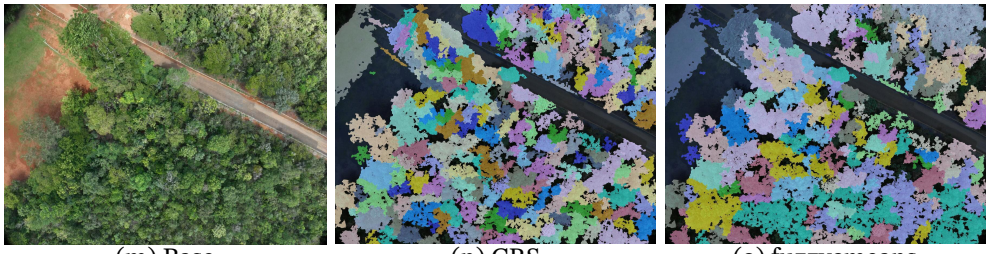
(d) Base (e) ETPS (f) fuzzycmeans



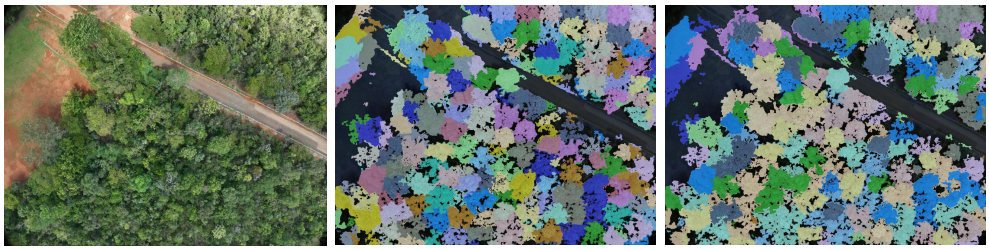
(g) Base (h) SEEDS (i) fuzzycmeans



(j) Base (k) ERS (l) fuzzycmeans



(m) Base (n) CRS (o) fuzzycmeans

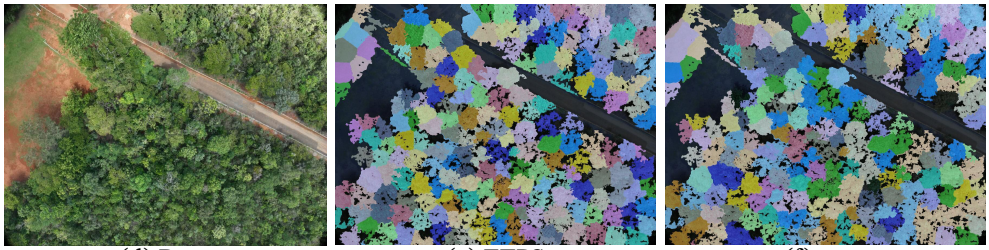


Fonte: Elaborada pelo autor

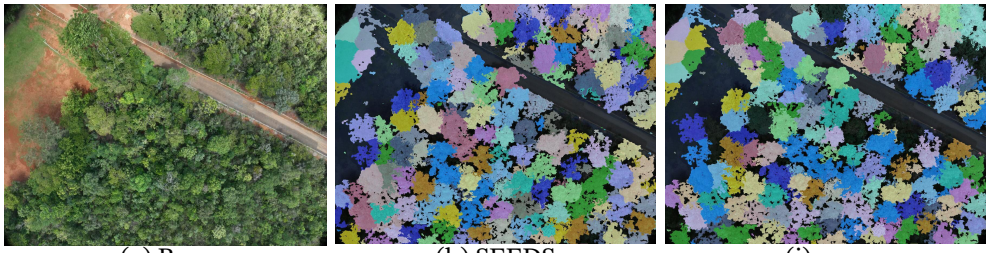
Apêndice J – SOM

Figura J.1 – Segmentos com cada segmentador clusterizados com som

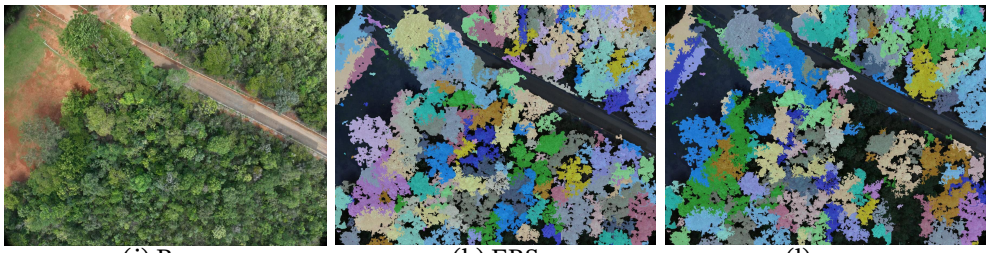
(a) Base (b) SLIC (c) som



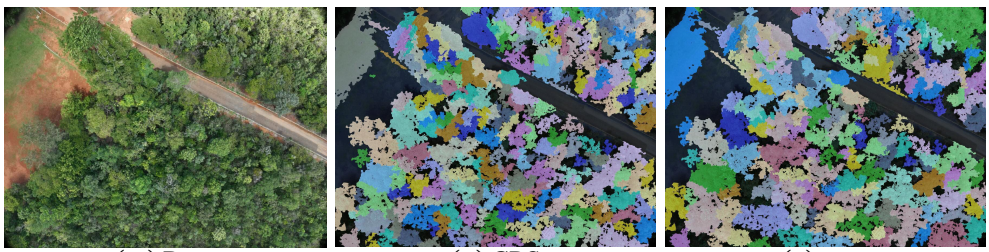
(d) Base (e) ETPS (f) som



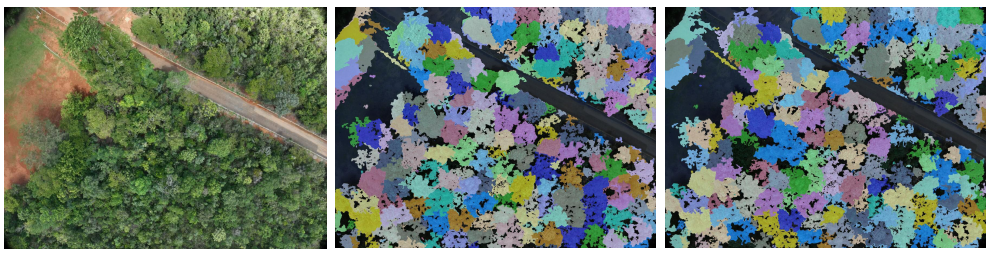
(g) Base (h) SEEDS (i) som



(j) Base (k) ERS (l) som



(m) Base (n) CRS (o) som



Fonte: Elaborada pelo autor


UnB